





**Méthode hybride pour la reconnaissance d'activités de la vie quotidienne et  
d'exercices physiques en temps réel**

**KÉVIN CHAPRON**

**Mémoire présenté à l'Université du Québec à Chicoutimi en vue de l'obtention  
du grade de la Maîtrise (M. Sc.) en informatique, profil Recherche**

Québec, Canada

© Kevin Chapron, 2017

## RÉSUMÉ

Dans le monde de la réadaptation, un souhait commun est d'utiliser un programme de plusieurs exercices physiques pour diminuer les troubles reliés à une maladie musculaire. C'est en particulier vrai pour les patients concernés par les maladies neuromusculaires. Ce programme d'exercices à réaliser chez soi est envisagé en moyen terme par les thérapeutes, étant donné que les patients atteints de maladies neuromusculaires ont de grandes contraintes de déplacement. Cependant, le thérapeute n'a aucune preuve que le patient réalise ses exercices chez lui. Pour se faire, la technologie peut reconnaître des activités, et notamment les exercices physiques.

De nouveaux capteurs voient le jour, de moins en moins chers et de plus en plus performants. Ces capteurs sont entre autres des accéléromètres, des gyroscopes, des magnétomètres. Ils sont très puissants et très utilisés pour confectionner des montres, des téléphones intelligents, des bracelets et des ceintures.

Pour permettre de reconnaître les exercices physiques prescrits par le thérapeute, et pour pouvoir quantifier le niveau d'exercice global d'un patient, une solution algorithmique a été développée à l'aide de matériel existant. Cette solution est composée de plusieurs périphériques communiquant entre eux par les moyens de protocoles sans fil. Les résultats obtenus par cette méthode de reconnaissance hybride sont excellents. De plus, malgré qu'elle ait été développée pour une population atteinte de dystrophie myotonique de type 1, elle peut être utilisée potentiellement pour différents types de patients.

## TABLE DES MATIÈRES

<b>RÉSUMÉ.....</b>	<b>III</b>
<b>TABLE DES MATIÈRES.....</b>	<b>IV</b>
<b>LISTE DES TABLEAUX.....</b>	<b>VI</b>
<b>LISTE DES FIGURES .....</b>	<b>VII</b>
<b>LISTE DES ÉQUATIONS.....</b>	<b>IX</b>
<b>LISTE DES ACRONYMES.....</b>	<b>XII</b>
<b>DÉDICACE .....</b>	<b>XIII</b>
<b>REMERCIEMENTS .....</b>	<b>XIV</b>
<b>CHAPITRE 1.....</b>	<b>1</b>
<b>INTRODUCTION .....</b>	<b>1</b>
1.1 CONTEXTE.....	1
1.2 PROBLÉMATIQUE CIBLÉE DANS CE MÉMOIRE .....	5
1.3 CONTRIBUTION DE LA RECHERCHE .....	7
1.4 MÉTHODOLOGIE DE LA RECHERCHE .....	8
1.5 ORGANISATION DU MÉMOIRE .....	9
<b>CHAPITRE 2.....</b>	<b>12</b>
<b>REVUE DE LITTÉRATURE .....</b>	<b>12</b>
2.1 DÉFINITIONS .....	12
2.1.1 FENÊTRAGE .....	13
2.1.2 POSITIONNEMENT .....	14
2.1.3 TAUX D'ACQUISITION DES DONNÉES .....	15
2.2 ALGORITHMES D'APPRENTISSAGE .....	15
2.2.1 CLASSIFICATION NAÏVE BAYÉSIENNE.....	17
2.2.2 RECHERCHE DES PLUS PROCHES VOISINS .....	20
2.2.3 ARBRES DE DÉCISION .....	22
2.2.4 FORÊTS D'ARBRES DÉCISIONNELS .....	25
2.2.5 MACHINES À VECTEURS DE SUPPORT.....	27
2.2.6 RÉSEAUX DE NEURONES ARTIFICIELS.....	28
2.3 CAS APPLIQUÉS .....	31
2.3.1 TRAVAUX EN LIEN AVEC LA RECONNAISSANCE D'AVQ .....	31
2.3.2 TRAVAUX EN LIEN AVEC LA RECONNAISSANCE D'EXERCICES PHYSIQUES	35

2.3.3 CONCLUSION .....	38
<b>CHAPITRE 3.....</b>	<b>41</b>
<b>PROTOTYPE.....</b>	<b>41</b>
3.1 INTRODUCTION.....	41
3.2 MATÉRIEL .....	42
3.2.1 BRACELET – INTEL EDISON .....	42
3.2.2 RASPBERRY PI 3.....	46
3.3 COMMUNICATION .....	47
3.4 VERSION 2.0 DU BRACELET.....	52
<b>CHAPITRE 4.....</b>	<b>55</b>
<b>MODÈLE DE RECONNAISSANCE HYBRIDE.....</b>	<b>55</b>
4.1 INTRODUCTION.....	55
4.2 PRÉSENTATION DES AVQ ET DES EXERCICES PHYSIQUES .....	55
4.3 ATTRIBUTS UTILISÉS .....	58
4.4 MÉTHODE DE RECONNAISSANCE HYBRIDE .....	64
<b>CHAPITRE 5.....</b>	<b>69</b>
<b>VALIDATION DU SYSTÈME .....</b>	<b>69</b>
5.1 INTRODUCTION.....	69
5.2 STRUCTURE DU JEU DE DONNÉES .....	69
5.3 ANALYSE DES RÉSULTATS.....	70
5.3.1 INDICATEURS DE PERFORMANCE DE CLASSIFICATION.....	70
5.3.2 RÉSULTATS .....	73
5.4 CONCLUSION .....	78
<b>CONCLUSION.....</b>	<b>80</b>
<b>RÉFÉRENCES .....</b>	<b>83</b>

## LISTE DES TABLEAUX

TABLEAU 1 - JEU DE DONNÉES POUR LA CLASSIFICATION NAÏVE BAYÉSIENNE .....	18
TABLEAU 2 - DESCRIPTIF DES BLOCS UTILISÉS AVEC L'EDISON.....	46
TABLEAU 3 - DESCRIPTION DES EXERCICES PHYSIQUES ÉTUDIÉS.....	57
TABLEAU 4 – DÉFINITIONS MATHÉMATIQUES DES ATTRIBUTS DU DOMAINE TEMPOREL.....	59
TABLEAU 5 - DÉFINITIONS MATHÉMATIQUES DES ATTRIBUTS DU DOMAINE FRÉQUENTIEL .....	63
TABLEAU 6 - MATRICE DE CONFUSION POUR LE PARADOXE DE L'EXACTITUDE, PARTIE 1 .....	71
TABLEAU 7 - MATRICE DE CONFUSION POUR LE PARADOXE DE L'EXACTITUDE, PARTIE 2 .....	71
TABLEAU 8 - DESCRIPTION DES INDICATEURS DE PERFORMANCES DE CLASSIFICATION.....	72
TABLEAU 9 - CATÉGORIES D'ALGORITHME ET LEUR REPRÉSENTANT .....	77

## LISTE DES FIGURES

FIGURE 1 - EXEMPLE DE FENÊTRE DE CHEVAUCHEMENT À 50%.....	13
FIGURE 2 - EXEMPLE DE K-NN .....	20
FIGURE 3 - EXEMPLE D'APPLICATION DE FORÊTS D'ARBRES DÉCISIONNELS .....	26
FIGURE 4 - EXEMPLE D'UNE MACHINE À VECTEURS DE SUPPORT .....	27
FIGURE 5 – EXEMPLE DE PERCEPTRON MONOCOUCHE SIMPLE .....	29
FIGURE 6 - EXEMPLE DE PERCEPTRON MULTICOUCHE .....	30
FIGURE 7 - APPAREILS MOBILES DÉTENUS, SELON L'ÂGE (CEFRIO, 2015) .....	39
FIGURE 8 - DISPOSITIF INTEL EDISON SEUL.....	42
FIGURE 9 – EXEMPLE D'ACCÉLÉRATIONS COLLECTÉES LORS DE LA RÉALISATION D'UNE ACTIVITÉ (SQUAT SUR CHAISE) .....	43
FIGURE 10 – EXEMPLE DE VARIATIONS COLLECTÉES AVEC UN GYROSCOPE LORS DE LA RÉALISATION D'UNE ACTIVITÉ (SQUAT SUR CHAISE).....	44
FIGURE 11 - DISPOSITIF INTEL EDISON ASSEMBLÉ AVEC D'AUTRES BLOCS .....	45
FIGURE 12 – EXEMPLE DE VARIATIONS COLLECTÉES AVEC UN MAGNÉTOMÈTRE LORS DE LA RÉALISATION D'UNE ACTIVITÉ (SQUAT SUR CHAISE) .....	45
FIGURE 13 - RASPBERRY PI 3.....	47
FIGURE 14 – FONCTIONNEMENT DU PROTOCOLE TCP .....	48
FIGURE 15 - CONSOMMATION D'ÉNERGIE DES DIFFÉRENTS PROTOCOLES (LEE <i>ET</i> <i>AL.</i> , 2007), P6 .....	51

FIGURE 16 - CARTE DE DÉVELOPPEMENT DU NRF52832 .....	52
FIGURE 17 - EXERCICES PHYSIQUES ÉTUDIÉS.....	56
FIGURE 18 - PSEUDO-CODE DE LA MÉTHODE DE RECONNAISSANCE HYBRIDE.....	67
FIGURE 19 - SCHÉMA REPRÉSENTANT LE FONCTIONNEMENT DE LA MÉTHODE DE RECONNAISSANCE HYBRIDE .....	68
FIGURE 20 - REPRÉSENTATION D'UNE VALIDATION CROISÉE AVEC $k = 4$ .....	73
FIGURE 21 - RÉSULTATS PROPRES POUR LES AVQ (FORÊTS D'ARBRES DÉCISIONNELS) .....	74
FIGURE 22 - RÉSULTATS PROPRES POUR LES EXERCICES PHYSIQUES (FORÊTS D'ARBRES DÉCISIONNELS) .....	75
FIGURE 23 - ANALYSE EN LOO DES AVQ (FORÊTS D'ARBRES DÉCISIONNELS) .....	76
FIGURE 24 - ANALYSE EN LOO DES EXERCICES PHYSIQUES (FORÊTS D'ARBRES DÉCISIONNELS) .....	77
FIGURE 25 - COMPARAISON DES ALGORITHMES D'APPRENTISSAGE .....	78



## LISTE DES ÉQUATIONS

ÉQUATION 1 - DISTRIBUTION DE PROBABILITÉS GAUSSIENNES .....	17
ÉQUATION 2 - ÉQUATION DE LA CLASSIFICATION NAÏVE BAYÉSIENNE .....	17
ÉQUATION 3 - EXEMPLE DE CALCUL DE L'ESPÉRANCE STATISTIQUE .....	19
ÉQUATION 4 - EXEMPLE DE CALCUL DE LA VARIANCE .....	19
ÉQUATION 5 - EXEMPLE DE CALCUL DE LA DISTRIBUTION DE PROBABILITÉ GAUSSIENNE .....	19
ÉQUATION 6 - DISTANCE DE MANHATTAN .....	21
ÉQUATION 7 - DISTANCE EUCLIDIENNE .....	21
ÉQUATION 8 - ENTROPIE DE SHANNON .....	22
ÉQUATION 9 - ENTROPIE DE SHANNON POUR LES ATTRIBUTS SEMBLABLES .....	23
ÉQUATION 10 - GAIN D'INFORMATION.....	23
ÉQUATION 11 - INDICE DE DIVERSITÉ DE GINI .....	24
ÉQUATION 12 - NOMBRE DE CARACTÉRISTIQUES OPTIMAL POUR UN REGROUPEMENT.....	27
ÉQUATION 13 – FONCTION SIGN.....	30
ÉQUATION 14 - CALCUL DE LA MOYENNE POUR UN AXE DONNÉ .....	60
ÉQUATION 15 - CALCUL DE LA MOYENNE DES MOYENNES DE CHAQUE AXE .....	60
ÉQUATION 16 - CALCUL DE L'ÉCART-TYPE POUR UN AXE DONNÉ .....	60

ÉQUATION 17 - CALCUL DE LA MOYENNE DES ÉCART-TYPES DE CHAQUE AXE .....	60
ÉQUATION 18 - CALCUL DE SKEWNESS POUR UN AXE DONNÉ .....	60
ÉQUATION 19 - CALCUL DE LA MOYENNE DES SKEWNESS POUR CHAQUE AXE .....	60
ÉQUATION 20 - CALCUL DE KURTOSIS POUR UN AXE DONNÉ .....	61
ÉQUATION 21 - CALCUL DE LA MOYENNE DES KURTOSIS POUR CHAQUE AXE .....	61
ÉQUATION 22 - CALCUL DU ZCR SUR UN AXE DONNÉ .....	61
ÉQUATION 23 - CALCUL DE LA MOYENNE DES ZCR POUR CHAQUE AXE .....	61
ÉQUATION 24 - CALCUL DE LA CORRÉLATION ENTRE DEUX AXES .....	61
ÉQUATION 25 - CALCUL DE LA COVARIANCE ENTRE DEUX AXES .....	62
ÉQUATION 26 - CALCUL DE L'ÉNERGIE .....	63
ÉQUATION 27 - CALCUL DE L'ENTROPIE .....	63
ÉQUATION 28 - SOUS-CALCUL NÉCESSAIRE POUR LE CALCUL DE L'ÉNERGIE .....	63
ÉQUATION 29 - CALCUL DU COEFFICIENT DC .....	63
ÉQUATION 30 - CALCUL DE L'EXACTITUDE .....	70
ÉQUATION 31 - CALCUL DE L'EXACTITUDE POUR LE PARADOXE DE L'EXACTITUDE, PARTIE 1 .....	71
ÉQUATION 32 - CALCUL DE L'EXACTITUDE POUR LE PARADOXE DE L'EXACTITUDE, PARTIE 2 .....	71
ÉQUATION 33 - FORMULE POUR CALCULER LA PRÉCISION .....	72

ÉQUATION 34 - FORMULE POUR CALCULER LE RAPPEL .....	72
ÉQUATION 35 - FORMULE POUR CALCULER LE F-SCORE .....	72
ÉQUATION 36 - FORMULE POUR CALCULER LE KAPPA .....	73

## LISTE DES ACRONYMES

AVQ.....	Activités de la vie quotidienne
BLE .....	Bluetooth à basse consommation (Bluetooth Low Energy)
DM1.....	Dystrophie Myotonique de type 1
Edison .....	Bracelet Intel Edison
FN .....	Faux Négatif
FP.....	Faux Positif
G .....	Unité d'accélération (relative à la gravité)
Gauss.....	Unité électromagnétique d'induction magnétique
Hz.....	Hertz
I <sup>2</sup> C .....	Circuit Inter-Intégré (Inter-Integrated Circuit)
IA.....	Intelligence Artificielle
IAm.....	Intelligence Ambiante
k-NN.....	Algorithme des k plus proches voisins
LIARA.....	Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités
LiPo .....	Lithium-Ion Polymère
LOO.....	Leave-One-Out
SPI .....	Interface de périphérique série (Serial Peripheral Interface)
SVM .....	Machines à vecteurs de support
UART .....	Émetteur-récepteur asynchrone universel (Universal Asynchronous Receiver Transmitter)
VN .....	Vrai Négatif
VP .....	Vrai Positif
Wi-Fi.....	Ensemble de protocoles de communication sans fil
ZCR.....	Taux de passage par le zéro (Zero Crossing Rate)

## DÉDICACE

J'aimerais dédier ce mémoire à mes grands-parents : Jean-Claude Chapron et Jacqueline Chapron.

Tout d'abord, Mamie, tu as toujours été une source d'inspiration pour moi. Chaque fois que je réussissais quelque chose, je voyais la fierté dans tes yeux. J'espère que de voir ces quelques mots dans ce mémoire te provoquera le même bonheur que celui que tu m'as provoqué lorsque je t'ai vu me regarder à la collation des grades du baccalauréat. Tu es constamment dans mes pensées, toutes les lettres que tu m'envoies ne font que me conforter dans mon petit bout de pays à l'autre bout du monde.

Ensuite, Papi, tu es le roc que chacun devrait avoir. Malgré tout ce que tu as enduré dans la vie, tu es encore présent, souriant, profitant de la vie. Tu resteras toujours la personne la plus forte à mes yeux. Même si ton sourire n'apparaît pas souvent, je sais ce qu'il signifie lorsqu'il apparaît, et celui que tu me donnes quand je rentre du Québec pour venir te voir toi et mamie est incroyable.

Pour finir, vous êtes un couple comme on peut en voir très peu. Vous m'inspirez tous les deux dans ma vie, que ce soit dans vos comportements personnels, ou dans votre comportement amoureux. Même si vous n'arrêtez pas de vous lancer des vanes à tout bout de champ, ça se voit que vous vous aimez, et j'aimerais encore être comme ça quand j'aurai votre âge. J'espère vous faire plaisir du mieux que je peux avec ces quelques mots, et j'espère continuer à vous emplir de fierté.

À la meilleure grand-mère et au meilleur grand-père,

Je vous aime.

## REMERCIEMENTS

Pour commencer ce mémoire, je voudrais remercier l'ensemble des personnes qui m'ont soutenu pendant la réalisation de ces années de maîtrise. C'était un soutien indispensable à cette réussite.

Je remercie l'ensemble de l'équipe du LIARA (Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités) pour son soutien technique et financier. Plus particulièrement, je souhaite remercier les personnes suivantes : M. Bruno Bouchard pour avoir cru dans mes compétences et m'avoir permis d'intégrer l'équipe du LIARA en premier lieu, M. Sébastien Gaboury pour m'avoir encadré tout le long de ces années de maîtrise, pour son support, sa patience, sa compréhension et son expertise, et Mme Élise Duchesne pour ses idées et son support.

Je remercie également l'équipe présente au laboratoire pendant ces années de travail. Leur soutien a permis de trouver la motivation au travail, ou à se changer les idées lorsque l'envie n'était plus là. Plus particulièrement, je souhaiterais remercier Baptiste Lemarcis pour m'avoir permis de rentrer en contact avec M. Bouchard, ce qui m'a ensuite permis d'intégrer la maîtrise à l'UQAC.

Cependant, mes remerciements les plus chaleureux vont à ma famille, et plus particulièrement à mon père, M. Luc Chapron. Il a été présent à chaque grande étape de ma vie alors que nous vivons à plus de 5 000 km l'un de l'autre, que ce soit physiquement, financièrement ou moralement. L'ensemble de ma famille a été un support inimaginable, malgré le fait que j'ai décidé de réaliser mes études loin d'eux. C'est une étape dans ma vie qui me fait avancer à grands pas, et je les remercie vraiment d'accepter ma décision, et même de m'encourager à continuer dans ce sens.

Pour finir, je remercie également ma compagne, Sophie T. Bolduc pour ses encouragements continus et la patience dont elle dispose pour m'éprouver au quotidien. Son appui dans mes décisions est très important à mes yeux.

# CHAPITRE 1

## INTRODUCTION

### 1.1 CONTEXTE

La dystrophie myotonique de type 1 (DM1) est parmi les maladies neuromusculaires les plus fréquentes au monde. Elle possède son plus haut taux de prévalence au Québec, ce qui crée un environnement unique pour étudier sa complexité (Bouchard *et al.*, 1989). Cette maladie possède une hétérogénéité assez importante dans ses symptômes, comme des problèmes à la cataracte, une résistance à l'insuline et des maladies cardiovasculaires. Les muscles squelettiques sont aussi beaucoup touchés par cette maladie, comme démontré par la force musculaire maximale qui décline de 1 à 3% par an (Mathieu *et al.*, 2003). La perte progressive de force musculaire impacte la qualité de vie des patients en constituant un facteur prédictif fort de participation sociale perturbée (Gagnon *et al.*, 2008). De nos jours, aucun traitement n'est disponible afin de guérir ces patients : les recherches se concentrent alors sur des interventions permettant de diminuer la progression des déficiences liée à cette maladie. Pour ce faire, il a été démontré que le renforcement musculaire est une façon sécuritaire et efficace d'augmenter la force musculaire, et ainsi de diminuer la déficience musculaire. Cependant, la population DM1 est très sujette à l'apathie, comme l'indiquent les recherches de Gallais *et al.*, (2015), ce qui représente un facteur majeur qui pourrait compromettre la participation de cette population à un programme de renforcement musculaire assez demandant.

Les personnes atteintes par les maladies neuromusculaires, et en particulier la DM1, sont apathiques. Ce symptôme provoque une perte d'intérêt à l'égard de toute situation, ou de leur entourage. Ce manque d'intérêt accru entraîne une diminution de l'envie de réaliser des activités, qu'elles soient physiques ou sociales. Cette situation complique grandement l'implication de la population DM1 à compléter un programme d'exercices physiques spécialement conçu pour ce type de pathologie dans un centre de conditionnement physique

commercial. En effet, les difficultés de déplacement de cette population vers ce genre d'emplacements permettent de dire que l'entraînement à domicile serait une stratégie thérapeutique intéressante. En revanche, même si leur programme d'exercices s'effectue dans le confort de leur habitation, leur apathie posera les mêmes problèmes que dans une salle de conditionnement physique. Une solution offrant une forte probabilité de réussite serait de développer une technologie d'assistance adaptée à cette situation particulière. Elle devrait alors être capable de quantifier le niveau d'activité physique d'un patient à l'intérieur de son domicile, par exemple lorsqu'il marche, qu'il se lève, qu'il s'assoit. Les activités de ce type seront, tout au long de ce mémoire, désignées par activités de la vie quotidienne, ou AVQ. Mentionnons ici que le terme AVQ employé dans ce mémoire diffère de celui généralement rencontré en réadaptation ayant pour définition : « ensemble de gestes accomplis chaque jour par une personne dans le but de prendre soin d'elle-même ou de participer à la vie sociale ». De plus, elle devrait également pouvoir motiver le patient d'une manière personnalisée, afin de le guider dans l'exécution d'un programme de quelques exercices physiques (fente avant alternée, squat sur chaise) de renforcement des membres inférieurs prescrits par le thérapeute, et ce, à une certaine fréquence hebdomadaire. Évidemment, afin de pouvoir s'intégrer facilement à la demeure d'un patient, une telle technologie d'assistance doit être très peu coûteuse.

Un cas d'application lors d'une journée typique serait défini par la suite d'évènements suivante. Lorsque l'utilisateur se lève, il met en place le bracelet à son poignet. Tout au long de la journée, le bracelet acquiert les données et les transmet au nano-ordinateur afin d'effectuer la reconnaissance des activités de la vie quotidienne qu'il exécute et de quantifier son niveau d'activité global. De plus, lorsque l'utilisateur réalise son programme d'exercices, le nano-ordinateur reconnaît chaque exercice via le traitement des données reçues du bracelet afin de vérifier qu'il réalise le bon nombre de répétitions de ceux-ci. Si, lors de la journée, l'utilisateur n'a pas exécuté son programme d'exercices physiques, le nano-ordinateur le détectera et pourra émettre des encouragements pour qu'il l'entame. Lors du coucher,



l'utilisateur remet en charge le bracelet pour son utilisation du lendemain. Finalement, le nano-ordinateur compilera les différents résultats (quantification de l'activité physique quotidienne) et pourra ensuite générer un rapport qui sera transmis au professionnel traitant.

Heureusement, les progrès techniques et scientifiques récents permettent désormais d'envisager la conception et la création d'une telle technologie. Plus précisément, les différents dispositifs électroniques disponibles (comme les capteurs accélérométriques) rendent possible l'acquisition de données importantes produites par le patient lors de la réalisation de certaines activités. En combinant ces dispositifs avec les nouveaux algorithmes d'intelligence artificielle (IA) qui permettent d'analyser les données produites par ces dispositifs, il est possible de traduire des données ayant l'air abstraites en données intelligibles (par exemple, un mouvement particulier exécuté par le patient) pour effectuer une guidance, et procurer de l'assistance auprès du patient.

Cette solution potentielle s'inscrit directement dans la lignée des recherches sur l'intelligence ambiante (IAm). Ce sous-domaine de l'intelligence artificielle est un produit de l'informatique, qui remet en cause le concept même d'un ordinateur : le traitement s'effectue sur les interactions entre objets communicants (dispositifs électroniques) et humains. L'intelligence ambiante repose sur plusieurs composantes élémentaires qui peuvent être résumées par le principe de doter un environnement (ou une personne) d'une multitude de capteurs afin de collecter continuellement des données, et de pouvoir les analyser à l'aide d'algorithmes d'IA. L'IAm doit alors avoir des capteurs, ainsi que des dispositifs, choisis stratégiquement afin de les utiliser pour donner de l'assistance, tout en monitorant des activités grâce à l'IA. Le déploiement d'une telle technologie dans le foyer d'un patient lui permettrait de réaliser son programme d'entraînement (fourni par son thérapeute), tout en restant dans le confort de son foyer, permettant ainsi d'éviter les déplacements nécessaires vers un éventuel centre de conditionnement physique commercial.

Cependant, cette technologie devant pouvoir se déployer facilement dans le foyer d'un patient, elle doit être transportable aisément. Chaque personne n'a pas forcément les compétences technologiques pour installer et maintenir le bon fonctionnement d'une telle technologie chez elle, ce qui oblige la conception d'un tel système à être la plus simple possible. De plus, l'évolution des dispositifs électroniques de nos jours est tellement rapide que des capteurs de toutes tailles voient le jour. Étant donné la transportabilité désirée pour cette technologie, chaque partie doit être la plus petite possible, afin de ne pas être encombrante pour le patient, et par conséquent de ne pas permettre une baisse de motivation causée par la taille du système à utiliser.

Dans le cadre d'une technologie d'assistance, la composante d'intelligence du système repose, très souvent, sur la reconnaissance d'activités. C'est le processus permettant au système de convertir les données recueillies par les différents dispositifs vers une information intelligible, en utilisant des algorithmes d'IA. Ces algorithmes sont très souvent ceux du domaine du forage de données, mais ils peuvent également être utilisés, après modifications, dans le contexte de l'intelligence ambiante afin de faire de la reconnaissance d'activités en temps réel. Plusieurs dispositifs commerciaux permettent de quantifier le niveau d'activités quotidiennes, mais aucun dispositif n'est capable de reconnaître à la fois les AVQ ainsi que les exercices physiques lors d'un programme d'entraînement. De plus, les accéléromètres sont grandement utilisés comme moniteur du niveau d'activité physique dans les études en réadaptation. La technologie d'assistance à développer pour réaliser ce genre de dispositif est un réel défi, en raison des nombreuses conditions à respecter vues précédemment, et au respect de la personne sans perte de performance.

Afin de réaliser cette technologie d'assistance, il faut allier trois dispositifs distincts. Le premier dispositif est un périphérique de collecte de données de type accélérométrique, comme un bracelet. Le second est un périphérique de calcul et d'interprétation, afin d'acquérir par communication sans fil les données collectées par le premier périphérique et d'effectuer

l'ensemble des traitements (ex.: algorithmes d'intelligence artificielle). Ce périphérique est fondamental puisque le premier périphérique ne possède pas la puissance de calcul et l'énergie requise (le périphérique de collecte des données est alimenté par une batterie) pour réaliser ces traitements. Finalement, il faut également un moyen de pouvoir émettre des encouragements à l'utilisateur, par exemple par le biais d'un haut-parleur. Cependant, la conception et la création d'une technologie d'une telle envergure nécessiteraient beaucoup trop de temps, et par conséquent, les efforts ont été concentrés, dans le cadre de ce mémoire, sur les composantes primordiales à ce système: construire un périphérique de collecte de données ainsi que développer une nouvelle approche algorithmique afin de reconnaître deux types d'activités physiques (AVQ et exercices physiques prescrits par un thérapeute) en temps réel et d'incorporer cette approche novatrice au périphérique de calcul et d'interprétation vu précédemment.

## **1.2 PROBLÉMATIQUE CIBLÉE DANS CE MÉMOIRE**

Comme mentionné dans le contexte, pour créer la technologie d'assistance, il faut concevoir et créer deux périphériques : un premier qui permet de collecter des données accélérométriques en continu, et un autre qui permet d'analyser ces données, grâce à l'utilisation d'algorithmes d'IA.

Le premier périphérique est un dispositif qui doit être porté par la personne concernée, afin de récolter les données de ses mouvements. Cependant, ce dispositif ne doit pas être gênant, étant donné que la personne le portera continuellement (sauf pendant la nuit pour recharger la batterie). Une quelconque gêne dans l'utilisation de ce dispositif pourrait faire perdre l'intérêt du patient à l'utiliser. Il faut donc qu'il soit le plus naturel possible. Dans ce cadre, un grand périphérique serait à éviter, par exemple, à cause du fait qu'une grande taille pourrait empêcher son utilisateur de réaliser certains mouvements. De plus, le dispositif de collecte de données devant être porté par la personne, il faut que celui-ci soit sans fil, et par conséquent, autonome en énergie le plus longtemps possible. Pour ce faire, il doit avoir une

batterie lui distribuant l'énergie nécessaire à son fonctionnement et posséder des protocoles de communications sans fil efficaces et non énergivores.

Le second périphérique est un dispositif plus puissant qui permet de récolter les données émises par le premier périphérique et de les traiter. Afin de pouvoir recevoir les données, il faut qu'il suive également le même protocole de communications sans fil que l'autre dispositif. De plus, il doit être performant, étant donné qu'il va appliquer aux données des algorithmes de reconnaissance d'activités afin de pouvoir interpréter les données brutes et de les associer à la bonne activité physique. Une fois que les données ont été correctement associées au bon exercice physique, il est alors possible de quantifier les activités effectuées par la personne, que ce soit des AVQ ou des exercices physiques. Il serait également possible de prodiguer de l'assistance et de la guidance (ex.: développer un module algorithmique de messages vocaux diffusés par un haut-parleur), mais ce n'est pas la priorité de ce mémoire, comme cela a été énoncé auparavant.

Afin de reconnaître des activités à l'aide de données accélérométriques, il faut définir une fenêtre temporelle pendant laquelle l'activité se déroule. Cette fenêtre est le temps pendant lequel les données sont récoltées ; par exemple, si une fenêtre dure 3 secondes, les données recueillies pendant ces quelques secondes de données doivent correspondre à une activité précise. Les algorithmes de forage de données utilisent une seule fenêtre afin de reconnaître plusieurs activités : ces activités doivent alors toutes pouvoir être réalisées dans le même laps de temps. Cependant, dans le cadre de ce mémoire, les AVQ et les exercices physiques prescrits sont deux types d'activités bien distincts possédant des temps d'exécution très différents (ex. : exécuter une fente avant est une activité physique bien plus longue qu'une AVQ telle que se lever). L'approche algorithmique proposée doit alors permettre de reconnaître ces deux types d'activités, malgré que leur fenêtrage soit différent.

Le développement d'une telle approche basée sur les deux périphériques décrits ci-haut a alors plusieurs points clés. Tout d'abord, elle ne doit nécessiter en aucun temps l'utilisation d'un ordinateur chez le patient ; l'entièreté de la solution proposée doit être complètement autonome. Cependant, afin d'être très accessible, il faut que la solution proposée soit la moins dispendieuse possible, ce qui est rendu envisageable avec la quantité de nouveaux capteurs et de micro-ordinateurs voyant le jour de façon régulière. De plus, l'utilisation de ces nouvelles technologies au sein de cette solution permettra d'avoir un système relativement petit afin de pouvoir être déplacé facilement, et de pouvoir être déployé de manière très simple chez un patient.

### **1.3 CONTRIBUTION DE LA RECHERCHE**

Dans ce mémoire, trois contributions sont présentées afin de répondre à la problématique ciblée décrite à la section précédente. Ces contributions, une fois réunies, constituent une solution novatrice par rapport à la littérature en lien avec nos travaux.

Tout d'abord, aucune solution commerciale ne permettait de collecter les données et de les analyser simultanément. Afin de remédier à ce problème, un intergiciel a été conçu afin de pouvoir réunir un périphérique de collecte de données accélérométriques et un micro-ordinateur (Raspberry Pi3) capable d'analyser celles-ci en temps réel. Ce dispositif permet l'accès complet aux données brutes non filtrées (ce qui n'est généralement pas le cas avec les dispositifs commerciaux). Par ailleurs, il a aussi fallu choisir, implémenter et tester le bon protocole de communication entre ces deux périphériques (dispositif de collecte de données et le micro-ordinateur).

Ensuite, une solution algorithmique unique et novatrice capable de reconnaître deux types d'activités physiques (AVQ et quatre exercices physiques) a été développée. Cette solution a été testée et a démontré un taux de reconnaissance global de 96,50 %. Cette solution repose notamment sur les algorithmes d'apprentissage automatisé, d'intelligence

artificielle et l'analyse en temps réel des données accélérométriques sur des fenêtres temporelles de longueurs différentes. Par ailleurs, il est important de mentionner que cette solution algorithmique a fait l'objet d'une publication (évaluée par un comité de pairs) dans le compte-rendu de la conférence 14th IEEE International Conference on Ubiquitous Intelligence and Computing (Chapron *et al.*, 2017).

Pour terminer, une version améliorée du dispositif de collecte de données a été prototypée sans toutefois avoir été totalement assemblée et sérieusement testée. La quantité de travail qu'ont nécessitées les deux premières contributions justifie amplement le fait que ce nouveau prototype n'ait pas encore été testé. Par contre, les tests préliminaires indiquent que ce dernier, qui sera finalisé par l'équipe du LIARA, sera en mesure de remplacer efficacement la version initiale du dispositif de collecte de données. Deux grands changements sont suggérés pour cette nouvelle version : 1) le transfert de données via Bluetooth au lieu de Wi-Fi pour améliorer la consommation énergétique du prototype et 2) la conception et confection complète du prototype à partir de composants microélectroniques judicieusement choisies qui permettront de diminuer considérablement le coût du dispositif d'acquisition des données accélérométriques.

#### **1.4 MÉTHODOLOGIE DE LA RECHERCHE**

Le projet de recherche présenté dans ce mémoire a été réalisé en suivant une méthodologie de recherche divisée en quatre grandes étapes.

Tout d'abord, une phase indispensable est de savoir l'état actuel de la recherche dans ce domaine. Cette phase permet également d'acquérir les connaissances dans cette zone spécifique de la recherche. Afin de réaliser cette étape, une revue de littérature a été effectuée sur les différents types de reconnaissance d'activités physiques. Les différents moyens identifiés lors de cette revue ont alors été comparés afin de déterminer au mieux quelle façon de faire serait la plus performante dans le cadre de ce mémoire. Cette étape a été la plus

importante, car elle a permis d'envisager une solution viable pour répondre à la problématique émise, et de conduire à la contribution proposée de ce mémoire.

Ensuite, la prochaine phase était de comparer le matériel à utiliser pour les travaux. Ce choix était compliqué, en raison du fait qu'il allait être décisif dans l'avenir de ces travaux de recherche. Il fallait à la fois un ensemble le moins cher possible et assez puissant pour subvenir aux besoins de ce mémoire. Étant étudiant au LIARA, plusieurs dispositifs étaient disponibles, ce qui a permis d'avoir du choix parmi l'ensemble du matériel envisageable. De plus, il a fallu coupler l'ensemble des dispositifs disponibles avec les connaissances acquises lors de la revue de littérature pour avoir le meilleur dispositif possible pour cette tâche.

La troisième phase était la phase de développement. Il a fallu développer la solution pour effectuer la reconnaissance d'AVQ et d'exercices physiques sur des fenêtres de temps différentes avec un seul algorithme. Le développement de cette solution s'est effectué sur plusieurs dispositifs qui seront détaillés plus tard dans le chapitre 3. Ce travail a nécessité des connaissances dans le langage de programmation C++, et une connaissance approfondie du langage de programmation GoLang (appliqué pour créer la méthode de reconnaissance hybride). Ce choix technologique s'est porté sur la rapidité d'exécution qu'ils offraient.

La quatrième et dernière phase a permis de valider le système et sa programmation, en testant celui-ci en conditions « réelles ». Pour cette validation, un jeu de données du LIARA a été utilisé pour obtenir des résultats disponibles dans le chapitre 5 de ce mémoire. Une analyse approfondie de ces mêmes résultats est réalisée puis discutée, permettant de conclure à propos du travail effectué de reconnaissance hybride.

## **1.5 ORGANISATION DU MÉMOIRE**

Ce mémoire est composé de cinq chapitres. Le premier chapitre consiste en la description sommaire du projet, et permet d'introduire ce mémoire en tant que projet de

recherche. Tout d'abord, le contexte a été donné, suivi par la problématique maîtresse de ce mémoire. C'est la partie désignant la ligne directrice des travaux effectués.

Le second chapitre est la revue de littérature. Afin de rendre plus accessible ce mémoire, des définitions sont données à propos de termes ou de notions plus complexes. Ensuite, les algorithmes d'apprentissage utilisés dans ce domaine sont expliqués afin d'avoir une meilleure compréhension de la revue de littérature qui suivra. Cette même revue est séparée en deux catégories, à savoir les travaux en lien avec la reconnaissance d'AVQ, et ceux en lien avec la reconnaissance d'exercices physiques. Ce chapitre permettra d'avoir des informations sur l'implémentation du prototype à effectuer (positionnement, taux d'acquisition des données, etc.).

Le troisième chapitre détaille le système développé. Celui-ci est composé de plusieurs systèmes embarqués, et le détail de l'organisation entre eux est donné, afin de comprendre la mise en place conçue. Il débute avec le bracelet utilisé, puis avec le nano-ordinateur servant de puissance de calcul au système. Par la suite, l'ensemble des communications permettant au système de fonctionner sera expliqué, avec le cheminement nécessaire pour arriver à cette fin. De plus, un problème sur ce prototype sera exposé, ainsi que la conception permettant de résoudre le problème émis.

Le quatrième chapitre contient le modèle de reconnaissance hybride. Il est nommé « hybride » parce qu'il permet de reconnaître des AVQ et des exercices physiques sur des fenêtres de temps différentes. Une courte introduction sera donnée, suivie par une présentation détaillée des AVQ et des exercices physiques considérés dans ce mémoire. De plus, comme il le sera énoncé dans le chapitre 2, les algorithmes d'apprentissage ont besoin d'attributs pour effectuer une possible reconnaissance, ceux-ci seront alors détaillés dans ce quatrième chapitre. Pour finir, la méthode de reconnaissance hybride au complet sera



présentée, de même que le pseudo-code de celle-ci. Pour finir, un schéma illustrant la solution complète proposée sera donné.

Finalement, le cinquième chapitre consiste en la validation de l'outil et la présentation des résultats obtenus lors de celle-ci. Pour ce faire, un jeu de données disponible au LIARA, contenant les enregistrements d'une centrale inertielle acquis lors de la réalisation de diverses activités et exercices physiques a été utilisé. L'utilisation de ce jeu de données a permis de mesurer l'exactitude du système afin de démontrer sa validité. Pour finir, ce chapitre est suivi d'une conclusion générale de l'ensemble de ce mémoire.

## **CHAPITRE 2**

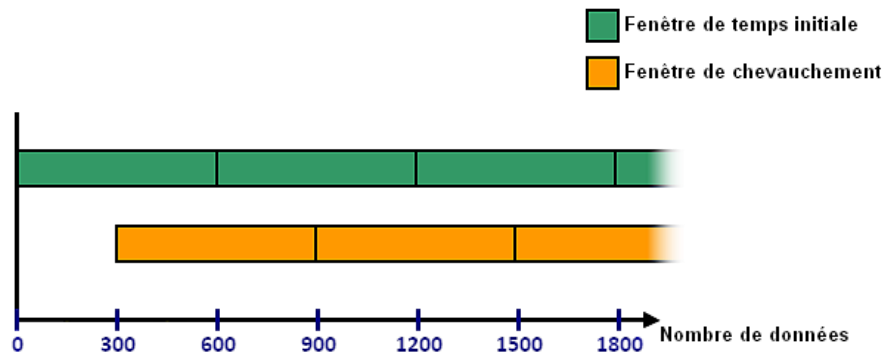
### **REVUE DE LITTÉRATURE**

Dans le domaine de la reconnaissance d'activités, beaucoup d'approches différentes existent afin d'accomplir cette difficile tâche. Dans le cadre de ce mémoire, les approches seront sélectionnées parmi celles traitant principalement avec des données accélérométriques, afin d'être en rapport avec les travaux effectués. Évidemment, il existe encore énormément d'autres approches qui sont utilisées avec des capteurs fournissant des données de nature différente, mais celles-ci ne seront pas abordées étant donné qu'elles ne sont pas en rapport direct avec la problématique de ce mémoire. Les approches sélectionnées vont varier en fonction du fenêtrage, de l'emplacement du dispositif, du nombre de dispositifs à utiliser, ainsi que du type de capteur exploité. Parallèlement au volet plutôt de nature matérielle, il y a une tâche fondamentale qui consiste à choisir les bons algorithmes à utiliser pour reconnaître les activités physiques désirées (AVQ et exercices physiques). Afin de rendre la revue de littérature claire, plusieurs définitions de concepts et de termes souvent utilisés seront données ainsi qu'une description détaillée des plus fréquents algorithmes employés dans le domaine de la reconnaissance d'activités physiques à partir de données accélérométriques. Pour conclure, une comparaison des plus récents travaux directement en lien avec la problématique de ce mémoire sera accomplie, afin de bien pouvoir positionner notre contribution scientifique au niveau théorique et pratique.

#### **2.1 DÉFINITIONS**

L'utilisation de la reconnaissance d'activités avec des données accélérométriques requiert plusieurs paramètres initiaux. Un paramètre indispensable afin que les techniques de forage de données fonctionnent correctement est celui du fenêtrage. Une brève description a été donnée dans le chapitre précédent, tandis qu'une description plus approfondie sera énoncée dans celui-ci. De plus, lorsque l'on utilise un capteur accélérométrique pour récolter

les données, il faut prendre en compte l'emplacement de celui-ci ainsi que son taux d'acquisition de données, et d'envoi de celles-ci.



**Figure 1 - Exemple de fenêtrage de chevauchement à 50%**

### 2.1.1 FENÊTRAGE

Le fenêtrage est un des paramètres les plus importants lors de la reconnaissance d'activités avec des données accélérométriques. C'est un intervalle de temps pendant lequel un utilisateur réalise une activité donnée. Par exemple, lorsque quelqu'un s'assoit sur une chaise (ceci est une AVQ), on peut considérer qu'il met deux secondes pour passer du stade « debout devant la chaise » à « assis sur la chaise ». Ces deux secondes sont alors le temps nécessaire afin de passer d'un stade à l'autre et représentent alors le fenêtrage de cette activité. C'est dans cet intervalle de temps qu'un algorithme de forage de données apprendra et reconnaîtra une seule activité. Cependant, en temps réel un problème arrive souvent. Une activité ne démarre pas forcément au départ du fenêtrage, en se finissant à la fin de celui-ci : il est possible d'avoir une activité chevauchant deux fenêtrages. Pour adapter le fenêtrage à ces cas, des fenêtrages chevauchants peuvent se rajouter aux fenêtrages initiaux. Celles-ci sont de la même longueur d'intervalle que les autres, mais utilisent un certain pourcentage de chevauchement. Ce pourcentage de données est la portion de la fenêtrage précédente qui est réutilisée. Par exemple, si une fenêtrage dure 6 secondes, et qu'une fenêtrage de chevauchement est paramétrée à 50% de ce temps, on aura alors une nouvelle fenêtrage se démarrant toutes

les 3 secondes. Dans les recherches de ce domaine, la valeur de chevauchement de 50% est souvent utilisée, puisqu'elle fournit des résultats pertinents (Bao, Intille, 2004; Ravi *et al.*, 2005; Cleland *et al.*, 2013; Mannini *et al.*, 2013). On peut voir à la Figure 1 un exemple de chevauchement paramétré à 50%. Dans le cadre de ce mémoire, deux fenêtres doivent être utilisées, une pour les AVQ, et une pour les exercices physiques. Une fois les paramètres de fenêtrage ajustés, il faut déterminer judicieusement le placement du dispositif d'acquisition de données.

### 2.1.2 POSITIONNEMENT

Lors de l'utilisation de capteurs accélérométriques, il faut avoir un placement corporel optimal. Les données accélérométriques sont très sensibles, ce qui permet d'avoir beaucoup de détails sur des mouvements, mais aussi des données indésirables. L'emplacement d'un tel capteur a alors beaucoup d'importance sur les données recueillies. Par exemple, lorsqu'un utilisateur réalise un mouvement de bras et que le capteur est positionné sur le pied gauche, les variations d'accélérations observées au niveau du pied sont pratiquement nulles, ce qui rend la reconnaissance d'activités très compliquée, voire impossible dans certains cas. Un placement stratégique du capteur doit alors être réfléchi afin d'avoir les données les plus fiables possible en fonction des activités à reconnaître. Dans le cadre de ce mémoire, l'étude est portée sur la quantification du niveau d'activités lors d'un renforcement musculaire, ce qui implique que la solution proposée doit savoir reconnaître à la fois les AVQ et les exercices physiques prescrits. Dans plusieurs recherches scientifiques comme celles de (Bieber *et al.*, 2010; Cleland *et al.*, 2013; Mannini *et al.*, 2013), un accéléromètre porté au poignet permet d'avoir des résultats très satisfaisants sur les AVQ. De plus, un dispositif porté à cet endroit est très envisageable pour toute population, étant donné que cet emplacement est souvent sollicité par la communauté par toutes sortes d'objets (montres, bijoux, bracelets). La sollicitation de cette partie du corps rend le travail avec celle-ci assez facile. Alors que les recherches se portent beaucoup sur les AVQ, ce mémoire traite également d'exercices physiques prescrits. Or, quoique ces exercices soient destinés à renforcer les jambes, ils font

intervenir des mouvements des bras assez discriminants pour les différencier, appuyant le choix d'utiliser le dispositif au poignet.

### **2.1.3 TAUX D'ACQUISITION DES DONNÉES**

Le dernier point à aborder est le taux d'acquisition des données. Ce taux est souvent exprimé en Hertz (Hz) ou en lectures par secondes. Théoriquement, plus le taux d'acquisition est grand, plus la reconnaissance va être précise. Cependant, avoir un dispositif capable de récolter plus de données et de les envoyer sur un réseau fermé est plus dispendieux que de récolter peu de données. Énormément de recherches portant sur la reconnaissance des AVQ utilisent un taux d'acquisition d'environ 20 Hz comme celles de Bao, Intille, (2004) ; Maurer *et al.*, (2006) ; Bieber *et al.*, (2010) ; Cleland *et al.*, (2013) ; Gao *et al.*, (2014), mais celles-ci ne se concentrent que sur des AVQ. Les exercices physiques sont plus complexes en termes d'exécution de mouvement. Si un mouvement plus rapide est effectué avec le même taux d'acquisition qu'un plus lent, celui qui est le plus rapide aura une perte de données à cause d'un taux d'acquisition trop bas. Il faut alors adapter cette valeur au plus haut possible pour reconnaître le mouvement le plus complexe. Par exemple, si un mouvement nécessite un taux d'acquisition de 80 Hz, et qu'un autre a besoin de 20 Hz, il faut alors calibrer le système sur 80 Hz. Lorsque l'ensemble des paramètres cités ci-dessus sont réunis, la prochaine étape pour réaliser la reconnaissance d'activités est l'utilisation d'algorithmes d'apprentissage. Il en existe énormément, c'est pourquoi les plus communs seront étudiés ci-après.

## **2.2 ALGORITHMES D'APPRENTISSAGE**

Pour reconnaître des activités, les algorithmes d'apprentissage sont très souvent utilisés. Il en existe plusieurs sortes, et ont tous leurs avantages et leurs inconvénients. Dans le domaine de la reconnaissance d'activités à l'aide de données accélérométriques, on va trouver toutes les sortes d'algorithmes, dépendant du type d'activités à détecter, et de leur nombre. Il faut donc évaluer quels algorithmes sont les plus pertinents au vu de l'étude de ce

mémoire, en comparant avec des études passées, et par le biais de tests effectués grâce à divers jeux de données. La comparaison avec les études passées permet alors d'avoir une idée des algorithmes à utiliser, tandis que les tests vont servir à vérifier le choix réalisé initialement. Elles vont aussi permettre de trouver les algorithmes les plus performants dans le cadre de ce mémoire. Il faut alors définir les algorithmes les plus utilisés, et pour ce faire, une description de chacun sera donnée, et des exemples de ceux-ci seront également exposés.

Les algorithmes d'apprentissage ont tous un principe de base en commun. Chaque algorithme apprend à reconnaître certaines activités. Pour permettre cet apprentissage, ils ont besoin d'un jeu de données initiales, regroupant des données significatives et propres : on parle d'ensemble d'apprentissages. Cet ensemble est constitué de plusieurs colonnes, et d'une classe. Les colonnes sont l'ensemble des attributs permettant de caractériser la classe qui, elle, désigne le nom de l'activité à reconnaître. Un algorithme d'apprentissage permet alors d'analyser ce jeu de données, et d'en extraire des caractéristiques, afin de pouvoir prédire une classe à un éventuel nouvel individu possédant les mêmes caractéristiques. Cependant, chaque algorithme a une façon distincte d'utiliser l'ensemble d'apprentissages, c'est pourquoi une description des algorithmes les plus utilisés dans ce domaine est fournie.

Tout d'abord, la classification naïve bayésienne sera abordée. C'est un algorithme très simpliste qui réussit à avoir de bons résultats sur certaines reconnaissances. Par la suite, l'algorithme consistant en la recherche des plus proches voisins sera examiné. Celui-ci ne fonctionne pas tout à fait comme les autres ; il compare un nouvel individu à classer avec l'ensemble d'apprentissages plutôt que d'extraire des caractéristiques de cet ensemble pour effectuer la classification. Cela peut engendrer un temps de reconnaissance relativement long, en fonction de la taille de l'ensemble d'apprentissages. Ensuite, les arbres de décisions seront passés en revue. Les arbres sont énormément utilisés, de par leur rapidité d'exécution, et les bons résultats qu'ils obtiennent. De plus, ces arbres sont également utilisés dans d'autres algorithmes, comme les forêts d'arbres décisionnels qui sont également présentées. Aussi, les

machines à vecteurs de support seront étudiées. Initialement, elles étaient utilisées afin de dissocier deux classes, mais des adaptations ont été réalisées afin qu'elles puissent permettre la reconnaissance de plus de deux classes différentes. Pour finir, les réseaux de neurones artificiels seront étudiés. C'est une méthode inspirée du système neuronal biologique.

### 2.2.1 CLASSIFICATION NAÏVE BAYÉSIENNE

La première approche étudiée est un type de classification probabiliste se basant sur le théorème de Bayes (Bayes, Price, 1763). Il s'agit d'un algorithme utilisant une méthode très simple permettant de pouvoir l'utiliser sur beaucoup de machines, même si celles-ci ne sont pas très performantes. Il existe plusieurs façons d'estimer les paramètres du modèle, mais il ne sera présenté ici que le modèle gaussien, étant donné que c'est celui qui est utilisé dans ce mémoire. Pour le respecter, il faut calculer l'espérance  $\mu_c$  et la variance  $\sigma_c^2$  pour chaque attribut de chaque classe dans l'ensemble donné. On peut alors supposer avec une observation  $v$ , qui permettrait d'avoir une probabilité de  $v$  sachant la classe  $c$ . Avec l'espérance et la variance calculée précédemment, il est possible d'écrire l'équation suivante pour obtenir la probabilité désirée :

$$p(x = v | c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}}$$

#### Équation 1 - Distribution de probabilités gaussiennes

De plus, grâce au théorème de Bayes et à l'équation ci-dessus, il est possible de déterminer le modèle probabiliste. Ce modèle permet alors de trouver la probabilité pour l'attribut sélectionné, en fonction d'une classe sélectionnée. Les attributs sont alors représentés par la lettre  $F$ , et la classe par la lettre  $c$ . La formule permettant alors de trouver la probabilité que ce soit la classe  $c$  en fonction des attributs  $F_1, \dots, F_n$  est la suivante :

$$p(c|F_1, \dots, F_n) = \frac{p(c) p(F_1, \dots, F_n | c)}{p(F_1, \dots, F_n)}$$

#### Équation 2 - Équation de la classification naïve bayésienne

Pour faciliter la compréhension de l'équation, elle peut être divisée en quatre parties représentées par les couleurs rouges, bleues, vertes et orange. La partie rouge représente la probabilité postérieure que la classe  $c$  soit observée lorsque le théorème de Bayes est appliqué. La partie bleue est la probabilité antérieure pour que la classe  $c$  soit vérifiée. La partie verte représente la vraisemblance : c'est l'ensemble de facteurs représentant la classe qui permettrait de dire que la classe est vérifiée. La partie jaune est une constante de normalisation qui est aussi appelée évidence. C'est la somme des probabilités antérieures, multipliées par leurs colonnes respectives.

Les calculs afin de trouver les valeurs d'espérance et de variance pour chaque attribut de chaque classe permettent de composer un nouveau jeu de données. Celui-ci contient plusieurs lignes, chacune représentant les résultats obtenus pour une classe donnée. Ce nouvel ensemble possède alors plus de colonnes que le jeu de données initial, mais beaucoup moins de lignes : c'est la vraisemblance. Lorsque ce nouvel ensemble est complété, il faut trouver la probabilité antérieure (partie bleue). Pour ce faire, il suffit de calculer le ratio entre le nombre d'échantillons de la classe par rapport à tous les échantillons.

Afin d'illustrer le développement de l'algorithme, un jeu de données très connu est utilisé. Il consiste en deux classes « masculin » et « féminin », et trois attributs : « Taille (cm) », « Poids (kg) » et « Pointure (cm) ». On peut l'observer dans le Tableau 1.

**Tableau 1 - Jeu de données pour la classification naïve bayésienne**

<b>Sexe</b>	<b>Taille (cm)</b>	<b>Poids (kg)</b>	<b>Pointure (cm)</b>
<i>masculin</i>	182	81.6	30
<i>masculin</i>	180	86.2	28
<i>masculin</i>	170	77.1	30
<i>masculin</i>	180	74.8	25
<i>féminin</i>	152	45.4	15
<i>féminin</i>	168	68.0	20
<i>féminin</i>	165	59.0	18
<i>féminin</i>	175	68.0	23



La première étape est alors de calculer l'espérance et la variance pour chaque attribut de chaque classe. Un exemple est fourni dans les équations suivantes pour chercher l'espérance et la variance de la taille pour la classe « masculin ». Dans chaque équation, N représente le nombre d'échantillons avec la classe donnée, et  $x_i$  la valeur d'un attribut de l'échantillon.

$$\mu_{c=\text{masculin}} = \frac{1}{N} \sum_{i=1}^N x_i = 178,$$

$$\sigma_{c=\text{masculin}}^2 = \frac{1}{(N-1)} \sum_{i=1}^N (x_i - \mu)^2 = 29,33.$$

**Équation 3 - Exemple de calcul de l'espérance statistique**

**Équation 4 - Exemple de calcul de la variance**

Une fois que l'on possède l'espérance et la variance pour chaque attribut de chaque classe, il faut trouver la probabilité antérieure. Étant donné que chaque classe a le même nombre d'occurrences, elles ont toutes les deux une probabilité antérieure de 0.5. À partir de ce moment, la classification est prête à recevoir des échantillons à classer. Pour continuer l'exemple, soit un échantillon de sexe inconnu, ayant 183 cm pour taille, 59 kg en poids et 20 cm en pointure. Avec ces nouvelles données, on peut calculer l'ensemble des valeurs nécessaires pour trouver la probabilité postérieure. De plus, comme l'évidence est une constante au dénominateur de la formule (voir partie jaune de l'équation 2), et que l'ensemble des valeurs est positif, celle-ci n'aura pas d'impact sur la décision finale. On peut alors uniquement comparer les numérateurs. L'équation suivante est un exemple de calcul menant au résultat final.

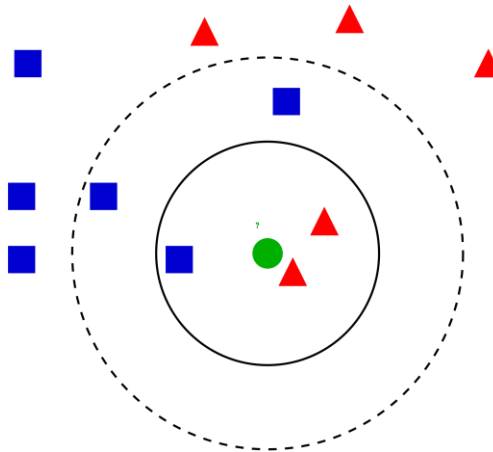
$$P(\text{taille}|\text{masculin}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(183-\mu)^2}{2\sigma^2}} = 4,8102 \times 10^{-2}.$$

**Équation 5 - Exemple de calcul de la distribution de probabilité gaussienne**

Lorsque chaque calcul est effectué, on peut alors calculer les numérateurs pour la classe « masculin », et pour la classe « féminin ». Les valeurs sont alors respectivement  $1,3404 \times 10^{-10}$ , et  $1,5200 \times 10^{-5}$ . On peut alors voir que la classe « féminin » possède une probabilité plus haute, l'échantillon est alors classé comme tel auprès de la classification naïve bayésienne.

### 2.2.2 RECHERCHE DES PLUS PROCHES VOISINS

Le second algorithme présenté est le *k-NN*, qui signifie « les k plus proches voisins » (Denœux, 1995). On peut alors séparer ce nom en deux : le nombre k, et le principe des plus proches voisins. Afin de mieux comprendre, la définition du principe sera énoncée en premier lieu, suivi d'une explication de l'impact du nombre k.



**Figure 2 - Exemple de k-NN**

La recherche des plus proches voisins est la base de cet algorithme. C'est une méthode très simple, mais très puissante à condition d'avoir un ensemble d'apprentissages assez imposant et avec le moins de bruit possible (le bruit est l'ensemble des valeurs non pertinentes). Cependant, cette méthode nécessite de comparer chaque individu du jeu de données initial, ce qui peut occasionner un rendement plus lent si l'ensemble d'apprentissages est trop grand. Afin de comparer un nouvel individu avec ceux du jeu de données, des mesures de distances entre ceux-ci sont calculées. Il existe beaucoup de mesure de distances

différentes, mais les plus rencontrées et les plus utilisées dans ce domaine sont la distance de Manhattan (Équation 6) et la distance euclidienne (Équation 7).

Soient  $A$  et  $B$  deux points dans un espace à  $N$  dimensions ayant respectivement comme coordonnées  $(x_1, \dots, x_N)$  et  $(y_1, \dots, y_N)$ . La distance de Manhattan entre ces deux points, notée  $d_M(A, B)$ , est calculée à l'aide de l'équation 6. La distance euclidienne entre ces deux points, notée  $d_E(A, B)$ , se calcule, elle, à l'aide de l'équation 7. Mentionnons que  $|x|$  représente la valeur absolue de  $x$ .

$$d_M(A, B) = \sum_{i=1}^N |x_i - y_i|$$

$$d_E(A, B) = \sqrt{\sum_{i=1}^N (x_i - y_i)^2}$$

**Équation 6 - Distance de Manhattan**

**Équation 7 - Distance euclidienne**

Ces deux distances occasionnent des résultats différents et nécessitent alors d'être essayées afin de vérifier quelle distance est la plus performante. Dans l'algorithme k-NN, ce calcul doit être effectué entre l'individu à classer et chaque individu de l'ensemble d'apprentissages. Une fois que chaque distance est obtenue, il suffit de comparer les plus proches voisins de ce nouvel individu. C'est à cet instant que le paramètre  $k$  entre en action. En effet, le nouvel individu sera classé en fonction des  $k$  plus proches voisins : la classe avec le plus haut nombre d'individus déterminera la classe du nouvel échantillon.

La valeur du nombre  $k$  est la pièce maîtresse de l'algorithme. Comme on peut le voir dans la Figure 2, si  $k=3$ , le nouvel échantillon sera classifié comme « triangle rouge », alors que si  $k=5$ , il sera classifié comme « carré bleu ». C'est également la raison pour laquelle il faut que l'ensemble d'apprentissages n'ait pas de bruit : si  $k$  a une valeur faible, et que du bruit est présent, il peut influencer la classification vers une fausse valeur.

### 2.2.3 ARBRES DE DÉCISION

Le troisième algorithme présenté est celui des arbres de décision. Leur structure rend leur interprétation très simple au premier coup d'œil, de par leurs nœuds, branches, et feuilles. Le passage d'un nœud à un autre s'effectue par un seuil appliqué sur une caractéristique discriminante. On peut alors parler de règle de classification. Ces règles doivent être écrites par l'algorithme qui va créer l'arbre. Il existe plusieurs types d'algorithmes comme ID3, C4.5, CART, et beaucoup d'autres. Les algorithmes C4.5, et CART seront expliqués afin d'indiquer plusieurs façons de créer des arbres de décisions.

#### 2.2.3.1 ID3

C4.5 et ID3 sont très semblables: ils ont été proposés par J. Quinlan (Quinlan, 1986). En 1993, J. Quinlan a publié un nouveau livre, afin de mettre à jour ID3 vers un nouvel algorithme plus performant : l'algorithme C4.5. Cependant, le principe de base reste le même. Ce principe consiste à créer l'arbre de décision récursivement. À chaque étape de la récursion, l'algorithme calcule pour chaque attribut, une valeur qui permettra de savoir lequel optimiserait le gain d'information. C'est ce qui permet de savoir quel attribut permettra de classer plus facilement à ce niveau de l'arbre. La valeur calculée est l'entropie de Shannon (Shannon, 1948), exprimée dans l'Équation 8. Dans la formule donnée,  $c$  est l'ensemble des  $k$  classes du jeu de données  $S$  noté  $c = \{c_1, c_2, \dots, c_k\}$ . De plus, l'ensemble des données qui possède  $N$  vecteurs de caractéristiques est noté  $T = \{X_1, X_2, \dots, X_N\}$ . De plus, chaque vecteur  $X_i$  possède un nombre  $n$  de caractéristiques noté  $X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,n}\}$ . Bien entendu, chaque  $X_i$  est associé à une classe  $c_j$ . Enfin, la fonction  $f(c_j, T)$  utilisée représente le nombre de fois que la classe  $c_j$  est existante dans  $T$ .

$$H(S) = - \sum_{j=1}^k \frac{f(c_j, T)}{N} \cdot \log_2 \frac{f(c_j, T)}{N}$$

Équation 8 - Entropie de Shannon

Lorsque l'entropie est calculée pour chaque attribut, il faut encore calculer une autre entropie, liée à la quantité d'attributs  $l$  semblables pour la même classe. Pour se faire, on utilise l'Équation 9 suivante.

$$H_l(T) = \sum_{l=1}^{N_l} \frac{f(x_{i,l}, T)}{N} \cdot \left( - \sum_{j=1}^k \frac{f(c_j, x_{i,l})}{N_l} \cdot \log_2 \frac{f(c_j, x_{i,l})}{N_l} \right)$$

#### Équation 9 - Entropie de Shannon pour les attributs semblables

Lorsque les deux calculs ont été réalisés correctement, il est possible de calculer la valeur du gain d'information  $G_l$ , qui est le gain d'information pour l'attribut  $l$ . La formule utilisée est donnée par l'Équation 10. Quand le gain d'information est calculé pour chaque attribut, l'attribut choisi est alors celui possédant le plus grand gain d'information. À ce moment, la scission sera effectuée sur l'attribut sélectionné. Cependant, il reste une dernière chose à vérifier, à savoir la valeur seuil à laquelle la scission doit s'effectuer. Une simple règle consiste en trier les données de l'attribut sélectionné de façon croissante, et d'en choisir la médiane. Cette valeur est alors la valeur de scission. Les deux sous-ensembles créés subissent le même traitement que vu précédemment, jusqu'à ce que les sous-ensembles ne possèdent qu'une seule classe.

$$G_l = H(S) - H_l(T)$$

#### Équation 10 - Gain d'Information

L'algorithme C4.5 possède quelques améliorations par rapport au principe de base d'ID3. Il permet l'utilisation de données continues et/ou discrètes. De plus, alors qu'ID3 n'autorise aucune donnée manquante, C4.5 permet d'avoir des attributs sans valeurs. Celles-ci sont alors ignorées, et ne sont pas prises en compte dans les calculs d'entropie. Une autre amélioration majeure de C4.5 est l'élagage (aussi appelé « *pruning* » en anglais) qu'il apporte

à l'arbre créé. En effet, il va repasser dans chaque branche de l'arbre afin de vérifier qu'elles apportent toutes une information discriminante et dans le cas contraire, il va essayer de remplacer la branche par une feuille résultante de cette branche sans pour autant perdre de données importantes.

### 2.2.3.2 CART

Le second algorithme étudié ici est celui de L. Breiman, nommé CART (Breiman *et al.*, 1984). Contrairement à ID3 et C4.5, celui-ci est basé sur un raisonnement binaire : à chaque noeud, il classifie la valeur du noeud, ou refait un arbre à partir de celui-ci afin de maximiser le gain d'informations. En effet, CART va classer récursivement chaque nœud en deux parties plus ou moins égales. Pour les équations suivantes, on va considérer la variable  $c$  qui dénote la même valeur que précédemment. De plus, on définit également l'ensemble  $M = \{c_1, \dots, c_N\}$ , qui définit l'ensemble des  $N$  classes possibles, ainsi que  $f_{c_i}$  étant la fraction des éléments de l'ensemble avec la classe  $c_i$ . L'algorithme prend alors le nœud de départ et donc l'ensemble des données initiales. Alors que ID3 ou C4.5 utilisent le gain d'information, CART utilise l'indice de diversité de Gini (Bishop *et al.*, 1975). C'est un indice mesurant la fréquence à laquelle un élément aléatoire serait mal classé si sa classe était sélectionnée aléatoirement depuis la distribution des classes dans le sous-ensemble. Avec l'ensemble des variables définies ci-dessus, on peut utiliser l'Équation 11 pour calculer l'indice de diversité de Gini.

$$I_G = 1 - \sum_{i=1}^N f_{c_i}^2$$

**Équation 11 - Indice de diversité de Gini**

C'est en minimisant la valeur de cet indice que CART choisit les deux sous-ensembles à venir dans l'arbre. Lorsque celle-ci est minimale, les deux sous-ensembles sont répartis sur la gauche et sur la droite de l'arbre, à partir du nœud courant. L'algorithme effectue alors

récurivement la même manœuvre sur chaque sous-ensemble jusqu'à avoir un arbre équilibré et complet.

#### **2.2.4 FORÊTS D'ARBRES DÉCISIONNELS**

L'algorithme des forêts d'arbres décisionnels a été conçu par L. Breiman (Breiman, 1999). Il consiste en l'utilisation de plusieurs arbres de classification, réunis ensemble. Par la suite, chaque arbre émet une hypothèse sur la classe du nouvel individu, et un vote est mis en place parmi chaque réponse. La classe la plus proposée est alors la classe déterminée par cet algorithme. Cependant, même si le principe est simple, il y a beaucoup de variables auxquelles il faut faire attention en utilisant cet algorithme.

La première se situe au niveau du nombre d'arbres à utiliser. En effet, théoriquement, plus il y a d'arbres et meilleure sera la classification. Cependant, avoir beaucoup d'arbres résulte en un temps d'apprentissage énormément plus long. Le nombre est en général trouvé par dichotomie en fonction des travaux réalisés, mais celles de (Krasotkina, Mottl, 2015) tendent à dire que le meilleur nombre d'arbres possible avant d'avoir un gain négligeable est de 128.

Une fois le nombre d'arbres défini, la prochaine étape est d'entraîner chaque arbre avec un jeu de données initial. Les arbres de décision sont très performants de par leur fonctionnement. Cependant, il est possible que ces arbres réalisent du surapprentissage : l'arbre de décision n'est alors capable de reconnaître que ce pour quoi il a été entraîné, et ne sera plus capable de reconnaître une classe alors qu'il devrait pouvoir. La solution à ceci proposée par l'auteur de l'algorithme est de modifier l'ensemble d'apprentissages, et de répartir celui-ci parmi tous les arbres de la forêt. Cela aura pour effet de réduire la variance des ensembles, pour permettre à l'algorithme de ne pas être surentraîné. Il permet également d'apporter à l'algorithme des parties de réponse, et donc d'ajouter une part d'aléatoire dans la

gestion des ensembles d'apprentissages. De plus, répartir l'ensemble d'apprentissages aura pour effet d'annuler les collections trop corrélées. Par exemple, si une classe est souvent établie par les arbres de décision en fonction d'une caractéristique spécifique et que celle-ci est retirée à quelques arbres, cela va forcer l'arbre en question à former un nouvel arbre, et donc une nouvelle façon de désigner une classe. Afin de réaliser cette séparation, il faut effectuer un regroupement dans les caractéristiques (« *features bagging* »). Soit  $\alpha$  le nombre d'attributs à garder dans chaque sous-ensemble d'apprentissage, et  $T$  l'ensemble des attributs noté  $T = \{X_1, X_2, \dots, X_N\}$ . Le fonctionnement du regroupement est alors d'effectuer  $\alpha$  tirages sans remise parmi la liste des caractéristiques de l'ensemble d'apprentissages. La classe reste associée aux  $\alpha$  attributs apportés par le regroupement. Soit  $A$  le nombre d'arbres dans la forêt, la valeur optimale pour  $\alpha$  est possible à obtenir grâce à l'Équation 12 ( $\lfloor x \rfloor$  signifie la partie entière de  $x$ ). Chaque sous-ensemble alors créé est envoyé à un arbre, afin qu'il apprenne à partir de ce nouvel ensemble. Ce regroupement de caractéristiques est effectué pour chaque

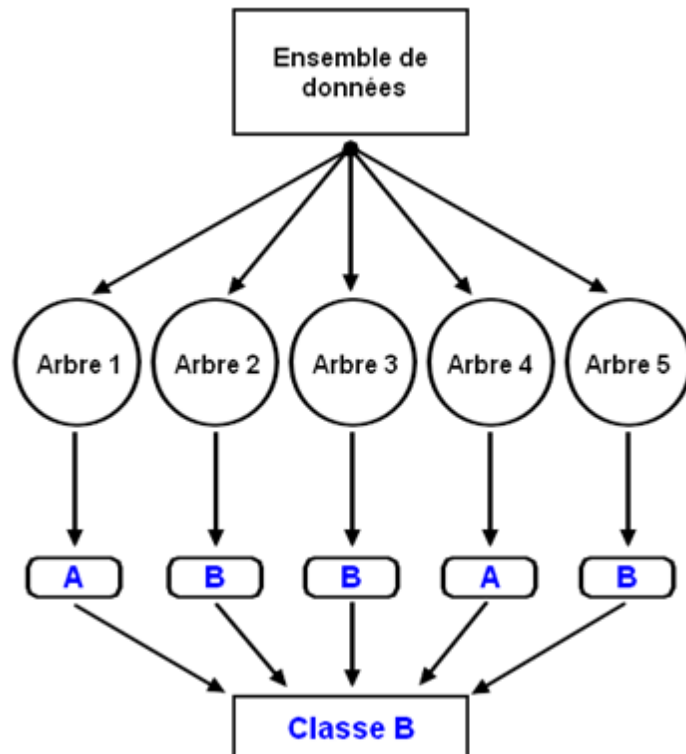


Figure 3 - Exemple d'application de forêts d'arbres décisionnels



arbre de la forêt, afin d'avoir une forêt initialisée avec des ensembles d'apprentissages différents. Un exemple de forêts d'arbres décisionnels est dans la figure 3.

$$\alpha = \lfloor \log_2 A + 1 \rfloor$$

### Équation 12 - Nombre de caractéristiques optimal pour un regroupement

Lorsque les arbres sont initialisés avec leurs ensembles, la classification d'un nouvel individu est généralement très rapide. Cette vitesse est directement reliée à celle des arbres de décisions étudiés précédemment. Les forêts d'arbres aléatoires sont un algorithme très utilisé en reconnaissance d'activités, de par ses excellents résultats et sa vitesse pour classer un nouvel échantillon. De plus, ceci le rend très attrayant pour une reconnaissance en temps réel.

#### 2.2.5 MACHINES À VECTEURS DE SUPPORT

Les machines à vecteurs de support (SVM) sont basées sur le principe de trouver un hyperplan afin de séparer deux classes distinctes. En effet, lorsque cet algorithme a été

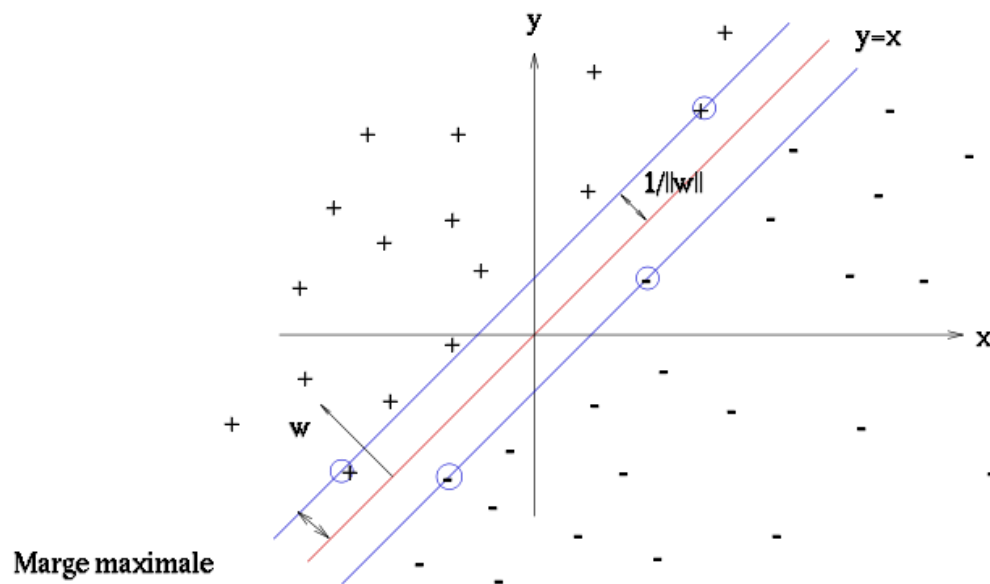


Figure 4 - Exemple d'une machine à vecteurs de support

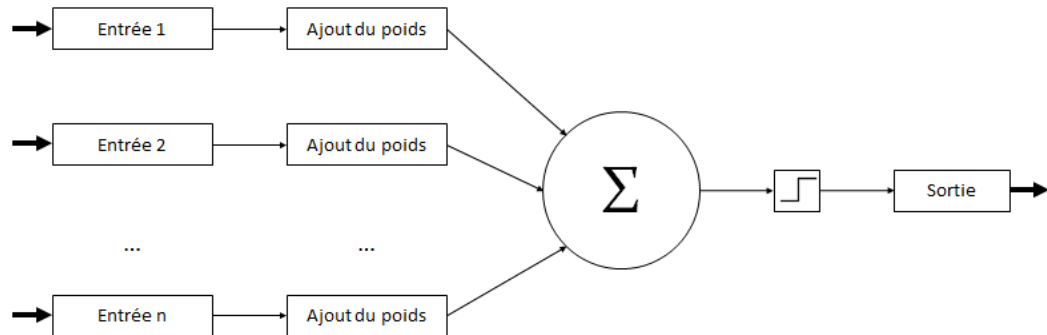
élaboré dans les années 90 par V. Vapnik (Vapnik, 1999), il avait été conçu pour distinguer seulement deux classes l'une de l'autre. Cependant, il existe une adaptation permettant à l'algorithme de reconnaître plus que deux classes.

L'algorithme permet de trouver l'hyperplan optimal séparant en deux les classes d'un ensemble avec la plus grande marge possible. Cette marge est alors calculée par la somme des distances entre les données les plus proches et l'hyperplan séparateur. Les données de chaque classe les plus proches de l'hyperplan sont alors sur un vecteur supplémentaire, qui s'appellera un vecteur de support. On peut voir dans la Figure 4 un exemple d'un hyperplan séparateur (ligne rouge) et de ses deux vecteurs de support (lignes bleues). Les marges maximales sont placées sur les points les plus proches, à savoir deux « + », et deux « - ». Les vecteurs de support doivent être équidistants par rapport à l'hyperplan séparateur. La classification d'un nouvel élément est alors très rapide, étant donné qu'il suffit d'appliquer une formule mathématique et de vérifier le positionnement du résultat par rapport à la classe. Cependant, la formule est rarement aussi simple à déterminer qu'avec l'exemple fourni. En effet, les jeux de données engendrent souvent des hyperplans sur plusieurs dimensions, rendant leur définition compliquée, et par conséquent le temps nécessaire à sa création important. Ce temps d'apprentissage peut être un handicap dans certains cas.

#### **2.2.6 RÉSEAUX DE NEURONES ARTIFICIELS**

La technique utilisant les réseaux de neurones artificiels repose sur une analyse d'un neurone biologique. En 1943, W. McCulloch introduit le concept, en tentant de modéliser de façon simplifiée les systèmes neuronaux humains (McCulloch, Pitts, 1943). Plusieurs améliorations ont été apportées par la suite, rendant ainsi, l'utilisation de cette méthode très populaire. Le concept d'un neurone formel est relativement relié à celui d'un neurone biologique : il y a un neurone artificiel, avec des entrées et une sortie. En outre, les neurones formels peuvent également posséder une interconnexion entre eux. Ces connexions

permettent à plusieurs neurones de résoudre des cas plus complexes, nécessitant plus d'un seul neurone.



**Figure 5 – Exemple de perceptron monocouche simple**

Il existe deux grands types d'architectures de réseaux de neurones. Cependant, étant donné l'utilisation qui est faite de ces algorithmes dans la littérature, il ne sera défini qu'une seule topologie qui est celle de la propagation vers l'avant. Ce terme indique que les informations ne peuvent circuler que dans une seule direction, à savoir de l'entrée jusqu'à la sortie. L'algorithme le plus utilisé est celui des perceptrons multicouches (« *MultiLayer Perceptrons* ») (Baum, 1988). Toutefois, afin de bien comprendre son fonctionnement, il est nécessaire d'examiner le fonctionnement d'un perceptron monocouche. Un exemple d'un perceptron monocouche est représenté à la Figure 5. Comme on peut le voir, il comporte plusieurs entrées. Chaque entrée est alors multipliée par un coefficient, qui représente le poids de cette entrée par rapport aux autres. Le résultat est contenu dans l'étape contenant le symbole  $\Sigma$ , qui en calcule la somme. Une fois cette somme récupérée, une fonction est appliquée à la valeur afin de catégoriser en deux classes distinctes. De plus, un seuil est soustrait à la somme préalablement calculée. Il existe beaucoup de fonctions différentes (ex.: linéaire, exponentielle, sigmoïde). Par exemple, il est possible d'appliquer une fonction *sign* telle que définie dans l'Équation 13.

$$\text{sign}(\gamma - \alpha) = \begin{cases} -1 & \text{si } \gamma - \alpha < 0 \\ 1 & \text{si } \gamma - \alpha \geq 0 \end{cases}$$

### Équation 13 – Fonction sign

Le principe du perceptron multicouche se base sur le même principe que le perceptron monocouche. Bien entendu, comme son nom l'indique, il possède plusieurs couches, alors que dans le perceptron unique, il y a une zone d'entrée et une zone de sortie directe. Lorsque les perceptrons sont connectés, le résultat de certains peut être utilisé dans l'entrée des autres. Un exemple d'un tel perceptron est montré la Figure 6. La grande différence ici est d'avoir plusieurs perceptrons. De plus, le concept de couche s'ajoute à l'innovation. Ces couches sont constituées de plusieurs perceptrons. On peut remarquer qu'il y a n couches, que la première contient les entrées et que la dernière contient les sorties. Cependant, entre ces deux couches spécifiques, il peut y avoir énormément d'étapes différentes consistant en l'ensemble des neurones permettant de résoudre des problèmes plus complexes. En effet, chaque neurone d'une couche est connecté avec chaque neurone de la couche suivante, ce qui permet un nombre conséquent d'enchaînements de fonctions linéaires. De plus, ces couches sont cachées. Le fait d'avoir ce genre de couches non-apparentes rend l'algorithme complètement opaque à l'utilisateur. Pour reconnaître des activités à l'aide de cet algorithme, il faut l'entraîner. Cependant, cet entraînement est excessivement long, mais permet généralement d'obtenir de bons résultats dans une grande majorité des problèmes de classification (Krishnan, Cook, 2014).

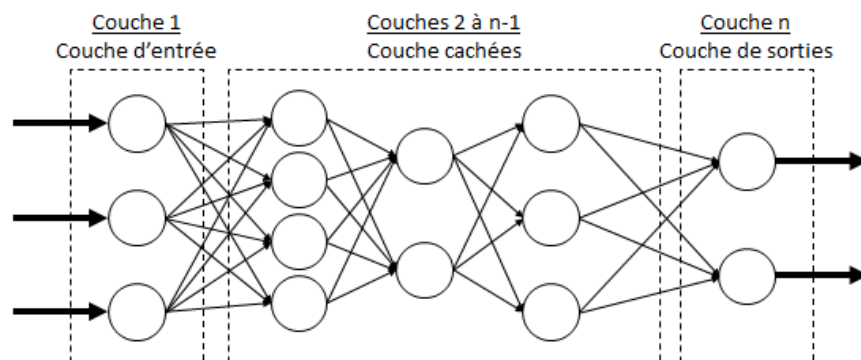


Figure 6 - Exemple de perceptron multicouche

## **2.3 CAS APPLIQUÉS**

Dans les précédentes sections, plusieurs termes ont été définis. De plus, les algorithmes les plus fréquemment utilisés ont été expliqués. Cependant, afin de positionner nos travaux par rapport aux travaux existants en relation directe avec notre problématique, une sélection de plusieurs articles a été effectuée. Dans un premier temps, les travaux menés sur de la reconnaissance d'AVQ seront étudiés. Étant donné que pour ce mémoire la reconnaissance va se porter à la fois sur les AVQ et les exercices, ceux-ci doivent également être étudiés dans une deuxième partie. Mentionnons que les attributs tels que la moyenne, la variance, l'énergie, l'entropie, la corrélation, le taux de passage par zéro qui sont énoncés dans les paragraphes qui suivent seront définis et utilisés (en partie) à la section 4 de ce mémoire.

### **2.3.1 TRAVAUX EN LIEN AVEC LA RECONNAISSANCE D'AVQ**

Beaucoup de recherches existent sur la reconnaissance d'AVQ grâce à des capteurs accélérométriques. Dans les paragraphes suivants, les travaux étudiés sont divisés, en trois catégories. Ces différentes catégories regroupent des travaux en fonction de leur mode de collecte des données.

La première catégorie de travaux est celle qui utilise des téléphones intelligents. Depuis plusieurs années, les téléphones intelligents deviennent plus populaires, de par leurs fonctionnalités. La masse d'outils disponibles est due à une quantité accrue de capteurs ajoutés, comme des accéléromètres, des gyroscopes ou des magnétomètres. En effet, l'analyse des données de l'accélération permet de réaliser des applications pour sauvegarder le matériel. Par exemple, Apple a utilisé ces capteurs afin d'arrêter le disque dur de ses périphériques pour le sauvegarder lors d'une chute. Beaucoup d'autres applications sont disponibles à partir de ce genre de capteurs dans des téléphones intelligents. Par exemple, les travaux de Kwapisz *et al.*, (2011) utilisent les accéléromètres afin de capturer les données, pour les analyser et identifier des activités simples. Ces activités se limitent aux AVQ connues,

à savoir marcher, courir, monter/descendre des marches, s'asseoir et se lever. Ils ont extrait six attributs différents, notamment la moyenne et l'écart-type sur une fenêtre de 10 secondes. Avec ces attributs, l'utilisation d'algorithmes d'apprentissage a été envisageable, et leurs résultats ont permis de dire que dans leur cadre spécifique d'études, le perceptron multicouche est le meilleur, avec un taux de reconnaissance de 91.7%. Ce taux est un rapport entre les classes prédites et celles censées être prédites.

Un autre cas d'études a reconnu ces mêmes activités, à savoir celles de Shoaib *et al.*, (2013). Cependant, dans leur cas, ils ont utilisé l'accéléromètre seul, le gyroscope seul, et leur combinaison afin de connaître l'impact de l'ajout de ces capteurs sur le taux de reconnaissance. Ils ont également essayé de positionner différemment le téléphone afin de voir quels effets sur le taux de reconnaissance cela pourrait entraîner. De plus, ils ont comparé beaucoup d'algorithmes connus, comme la classification naïve bayésienne, les machines à vecteurs de support, les réseaux de neurones, et autres algorithmes décrits dans la section 2.2 de ce mémoire. Ils ont formaté leurs données avec des fenêtres de deux secondes, et un pourcentage de fenêtre de chevauchement de 50%. Ils ont alors extrait deux attributs différents sur chacun des axes de leurs capteurs, à savoir la moyenne et l'écart-type, pour leur rapidité de calcul. Dans tous leurs tests, il est facile d'observer que la combinaison des capteurs accéléromètres/gyroscope est bien plus performante que ces mêmes capteurs seuls. Par contre, l'intégration d'un magnétomètre n'a pas été abordée dans leurs travaux. Les résultats qu'ils ont obtenus en positionnant le téléphone au poignet sont un peu moins performants qu'un positionnement à la ceinture ou dans la poche, mais s'avèrent quand même très performants pour chaque algorithme. En moyenne, l'algorithme *k-NN* est le plus performant, avec un taux de reconnaissance d'environ 97%.

Cependant, les recherches basées sur des téléphones intelligents sont loin d'être les seules dans leur domaine. Ces recherches étant relativement dispendieuses, dues au prix élevé d'acquisition et d'opération d'un téléphone intelligent, beaucoup de chercheurs se

tournent vers des solutions moins chères, avec l'utilisation directe de capteurs accélérométriques.

La seconde catégorie peut être abordée par l'approche réalisée dans (Gupta, Dallas, 2014), portant sur l'utilisation seule d'un accéléromètre à trois axes. Dans leurs travaux, ils ont reconnu des AVQ communes, comme marcher, courir, sauter, s'asseoir/se lever. Pour récolter les données, 7 sujets en bonne santé ont été volontaires. Les données collectées ont été analysées sur des fenêtres de 6 secondes, et un pourcentage de chevauchement de 50%. Néanmoins, en comparaison avec les approches précédentes, ils ont utilisé beaucoup d'attributs différents (11), répartis dans le domaine temporel et fréquentiel. Par exemple, ils se sont servis des attributs suivants : l'énergie, l'entropie, tout en gardant des attributs plus communs comme la moyenne, la variance et les tendances de celle-ci. Pour analyser leurs données, ils ont comparé deux algorithmes connus :  $k$ -NN (paramétré avec  $k=10$ ) et la classification naïve bayésienne. Leurs résultats sont très bons avec ces deux algorithmes : 98.4% comme taux de reconnaissance moyen pour  $k$ -NN, et 97.8% pour la classification naïve bayésienne.

D'autres travaux s'apparentent à celui-ci, comme ceux de Khan *et al.*, (2010), utilisant également juste un accéléromètre à trois axes. Ils reconnaissent également les AVQ standards, et n'ont utilisé que l'algorithme d'apprentissage des réseaux de neurones artificiels (perceptron multicouche). De plus, ils ont déterminé que leur fenêtre serait de 3.2 secondes, sans fenêtre de chevauchement. Leur expérimentation s'est déroulée à l'aide de 6 volontaires, lors d'un apprentissage non supervisé (chaque volontaire utilise alors le système d'enregistrement développé tout seul dans son environnement habituel). Les attributs extraits se résument à la moyenne, l'écart-type, l'entropie et la corrélation. En exécutant l'algorithme des perceptrons multicouches, ils obtiennent un taux de reconnaissance d'environ 97,65%.

Une dernière approche étudiée ici est celle de Mannini *et al.*, (2013). Leurs travaux portaient sur de la reconnaissance d'activités avec des données accélérométriques, où le capteur était situé soit au poignet, soit à la cheville. Afin de collecter leurs données, ils ont utilisé des fenêtres de 12.8 secondes sans chevauchement. Cependant, ils ont voulu comparer l'efficacité de cette longueur de fenêtre avec d'autres de 4 et 2 secondes. Ces deux dernières valeurs se sont révélées plus efficaces lors d'analyses en temps réel. Ils ont classifié 4 grandes classes définies par des sous-classes. Ces classes majeures sont « sédentaire », « cyclisme », « déplacement », « autres », et sont définies par les sous-activités qui se déroulent dans ces temps (par exemple, être assis, être debout, lire... définira la classe comme « sédentaire »). Ils ont obtenu de très bons résultats, résultant dans le fait que la position à la cheville augmente le taux de reconnaissance de 10.3% par rapport à la position du poignet. De plus, ils estiment que des fenêtres de l'ordre de 4 ou 2 secondes sont plus performantes avec de la classification en temps réel.

D'autres approches existent, et notamment celles utilisant plusieurs dispositifs simultanément. Afin d'introduire cette catégorie, les travaux de (Gao *et al.*, 2014) comparent les performances relevées d'un capteur unique et de multiples capteurs disséminés sur la personne. Pour atteindre ce résultat, ils ont collecté des données sur 8 personnes âgées (70 à 83 ans). Ces données ont alors été soumises à une fenêtre de 1 seconde sans chevauchement. De plus, ils ne se sont pas arrêtés à la problématique proposée, mais ont aussi étudié l'impact qu'avait ce changement en fonction des algorithmes d'apprentissage. Cinq algorithmes sont alors comparés : la classification naïve bayésienne, les réseaux de neurones artificiels, les arbres de décisions, les k plus proches voisins, et les machines à vecteurs de support. Pour appliquer ces algorithmes, ils ont calculé des attributs communs, auxquels ils ont ajouté des attributs plus complexes. Au total, 13 attributs différents ont été calculés, parmi lesquels se trouve la variance, le taux de passage par zéro (ZCR, nombre de fois que la valeur passe par la valeur 0), l'énergie, l'entropie ou encore la vélocité angulaire. Lorsque ces attributs sont calculés, chaque algorithme a été appliqué, et a permis de conclure



que les meilleurs algorithmes dans leur cas étaient les réseaux de neurones, les arbres de décisions et les k plus proches voisins, avec un taux de reconnaissance de 96%. En outre, leur recherche étant principalement sur la comparaison entre capteur unique et multicapteurs, ils ont démontré qu'utiliser plusieurs dispositifs avec un algorithme léger serait plus performant qu'un dispositif seul avec un algorithme de traitement plus performant (d'environ 2 à 10%). Un autre constat est que le temps requis pour entraîner les réseaux de neurones et les machines à vecteurs de support est significativement plus élevé que celui des autres algorithmes d'apprentissage (138 fois plus long qu'un arbre de décision pour les réseaux de neurones artificiels, et 17 fois plus long pour les machines à vecteurs de support). De plus, le temps requis à l'algorithme des k plus proches voisins pour effectuer la classification est colossal comparativement aux autres algorithmes (environ 80 fois plus long que la moyenne des autres algorithmes).

Plusieurs approches concernant la reconnaissance d'AVQ ont été abordées. Cependant, comme l'exige la problématique, les travaux liés à la reconnaissance d'exercices physiques doivent aussi être examinés.

### **2.3.2 TRAVAUX EN LIEN AVEC LA RECONNAISSANCE D'EXERCICES PHYSIQUES**

Alors que les travaux en lien avec la reconnaissance d'AVQ sont abondants dans cette catégorie de la recherche, ceux en lien avec la reconnaissance d'exercices physiques sont bien moins présents. Une sélection de quatre articles portant sur la reconnaissance d'exercices physiques ou d'activités plus précises que celles des AVQ a été effectuée et sera analysée ci-après. Dans la section précédente, l'analyse était séparée en trois catégories. Afin de réaliser une analyse cohérente, les mêmes catégories seront utilisées ici.

Tout d'abord, les travaux utilisant des téléphones intelligents pour reconnaître des exercices physiques spécifiques sont très peu nombreux. En effet, les travaux de (Bayat *et al.*, 2014) font partie des rares représentants de cette catégorie. Ils réalisent une reconnaissance

d'AVQ, mais reconnaissent également parmi ces activités basiques une activité de danse. Dans leur cadre de travail, cette activité de danser est considérée comme une AVQ, à savoir qu'elle est reconnue au même titre que les autres (marcher, courir, monter des marches, etc.). Ceci signifie que la reconnaissance de cette activité s'effectue avec les mêmes paramètres qu'une autre AVQ (ex.: le fenêtrage). Le fenêtrage qu'ils ont utilisé est de 128 enregistrements, à savoir 1,28 secondes étant donné que leur système possède une fréquence de 100 Hz. De plus, ils utilisent une fenêtre de chevauchement équivalente à 50% du fenêtrage initial. Leurs travaux comparent plusieurs algorithmes d'apprentissage sur un jeu de données qu'ils ont enregistré. Lorsque chaque résultat est obtenu pour chaque algorithme, il est facile de voir que l'algorithme des forêts d'arbres décisionnels est très performant (meilleur algorithme pour la reconnaissance de la danse lorsque le téléphone est dans la main). Cependant, pour avoir un meilleur taux de reconnaissance, ils ont utilisé une combinaison des résultats obtenus par trois algorithmes différents à savoir les forêts d'arbres décisionnels, les machines à vecteurs de support, et les réseaux de neurones artificiels. Leurs recherches permettent alors d'obtenir un taux de reconnaissance maximal de 91,15%.

La seconde catégorie analysée était celle utilisant un capteur accélérométrique unique. Parmi les travaux existants, un seul article mérite mention. En effet, la grande majorité des travaux présents dans ce domaine parlent des AVQ comme des activités physiques, ce qui complique énormément la recherche. L'analyse de cette catégorie se rapporte alors aux travaux de (Ravi *et al.*, 2005), qui permet de reconnaître des activités plus spécifiques que les AVQ basiques. Ils se sont servis d'un accéléromètre triaxial positionné au niveau de la région pelvienne, et envoyant ses données par la technologie Bluetooth à un taux de 50 Hz. Le fenêtrage choisi était alors de 256 enregistrements, représentant 5,12 secondes de données. La fenêtre de chevauchement utilisée était encore de 50% du temps initial. Ils ont extrait 4 attributs, dont trois du domaine temporel (moyenne, écart-type et corrélation), et un du domaine fréquentiel (énergie). Les activités qu'ils ont reconnues sont surtout du domaine des activités spécialisées. En effet, ils reconnaissent des AVQ standard mais également l'activité

de « passer l'aspirateur » ou encore de « se brosser les dents ». De même que pour les travaux observés précédemment, ils ont décidé de placer ces deux activités dans le même cadre que celles des AVQ, avec le même fenêtrage. Les meilleurs taux de reconnaissance obtenus sont ceux procurés par un algorithme de « *Plurality Voting* » qui ressemble en partie à celui des forêts d'arbres décisionnels, avec un taux de reconnaissance de plus de 90%. (Todorovski, Dzeroski, 2003)

La dernière catégorie est celle utilisant plusieurs capteurs à la fois pour reconnaître des exercices physiques. Cette catégorie possède plus de travaux essayant de reconnaître des exercices physiques, étant donné que la reconnaissance de ce type d'activité est plus simplifiée avec plusieurs capteurs (Gao *et al.*, 2014).

La première étude se porte sur les travaux de (Ellis *et al.*, 2014). Ils utilisent deux accéléromètres situés sur la hanche et le poignet, tous deux avec un taux de transmission de 30 Hz. Ils ont enregistré beaucoup d'activités différentes, comme des activités de ménage, du sport, du conditionnement physique, etc. Leurs travaux portaient sur la détection du comportement d'un utilisateur, pour pouvoir évaluer la quantité de temps qu'un utilisateur passait à s'entraîner par rapport aux positionnements sédentaires (« être assis » ou « être allongé »). Leur taux de reconnaissance moyen était de 85.6%. De plus, avec les données qu'ils ont recueillies, ils ont pu évaluer qu'une femme participante à leur expérimentation a passé plus de 74% de son temps en positions sédentaires. Pour reconnaître les différentes activités, les forêts d'arbres décisionnels ont été utilisées, tandis que pour les associations des activités avec un comportement, les modèles de Markov cachés ont été utilisés (Rabiner, Juang, 1986).

Un second article présentant les travaux de (Laerhoven *et al.*, 2012) est aussi intéressant. Ces derniers ont construit un harnais complet réunissant 30 accéléromètres ensemble, étant capables de communiquer ses axes à un taux de transmission de 30 Hz en

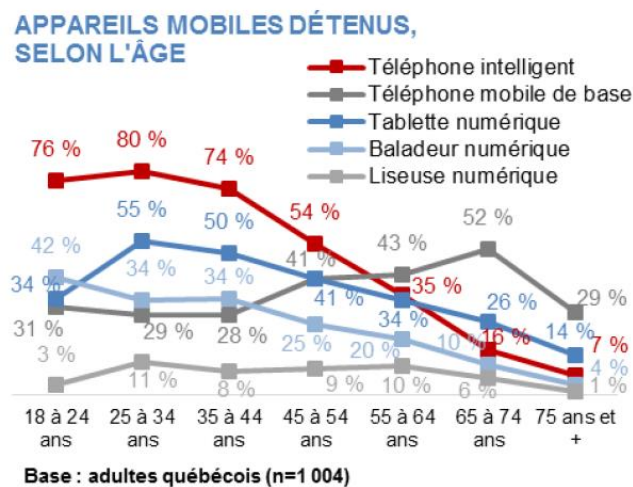
pire cas. Ils sont reliés en filaire à un microprocesseur qui garde les données en mémoire pour les analyser. Ils voulaient envoyer les données à un autre périphérique afin de les analyser en temps réel, mais l'utilisation d'autant d'accéléromètres a empêché une telle communication. Ils se sont alors focalisés sur la communication entre leurs capteurs afin d'optimiser les transferts de données. En effet, ils proposent de distribuer le formatage des données en fonction du nombre de capteurs présents sur le harnais, afin de pouvoir envoyer ensuite les données vers un périphérique capable de les analyser. Les activités qu'ils ont essayé de reconnaître étaient très précises : « s'agenouiller sur le genou droit », « sauter », « coup de genou gauche », « être assis avec les jambes croisées », et d'autres activités. Cependant, ils n'ont pas publié de résultat, déclarant que la performance d'un système dépendait du nombre de capteurs, et du contexte auxquels ils sont appliqués. De plus, ils ont réalisé cette étude en se focalisant sur une communication bien distribuée plutôt que sur un haut taux de reconnaissance.

Plusieurs approches ont été étudiées dans le domaine de la reconnaissance d'AVQ ou d'exercices plus spécifiques. Les travaux observés contiennent tous beaucoup d'informations qui sont intéressantes à utiliser pour répondre à la problématique de ce mémoire.

### **2.3.3 CONCLUSION**

Dans le cas de ce mémoire, les potentiels utilisateurs doivent avoir un système simple et qui ne décourage pas l'utilisation. En effet, la population ciblée manquant déjà d'intérêt à cause de leur apathie, il ne faut pas ajouter un manque de motivation à cause d'un système trop complexe. De ce fait, l'utilisation du système proposé doit être simple. Dans les travaux étudiés, beaucoup portaient sur l'utilisation de multiples capteurs afin de reconnaître plus distinctement les activités désirées. Cependant, demander à un patient apathique de réaliser son programme d'entraînement à l'aide d'un harnais comme dans les travaux de (Laerhoven *et al.*, 2012) est quelque chose d'inenvisable. De plus, il a pu être observé que le positionnement des capteurs au poignet était relativement efficace. Étant donné que la population est habituée à l'utilisation de leur poignet comme support (bracelets, montres), il est

beaucoup plus plausible d'utiliser cet emplacement comme point de départ pour le système. De plus, d'autres travaux utilisaient des téléphones intelligents. Cependant, la population ciblée ne possède pas forcément de telles technologies et n'a pas forcément les moyens de s'en procurer. En outre, malgré que ce projet soit basé sur une population DM1, une autre qui pourrait sûrement trouver profit d'un entraînement personnel est la population vieillissante. Comme le montre la figure 7, l'utilisation de téléphones intelligents auprès de cette population est loin d'être très populaire. La solution du bracelet est alors la plus plausible, et celle adoptée pour le projet.



**Figure 7 - Appareils mobiles détenus, selon l'âge (CEFRIQ, 2015)**

En ce qui concerne la reconnaissance en elle-même, beaucoup d'algorithmes sont utilisés. Le premier point à aborder est au niveau du type d'algorithme utilisé. Pour commencer, les forêts d'arbres décisionnels sont très populaires et permettent l'obtention de très bons résultats. Leurs utilisations dans les multiples travaux étudiés ont toujours permis d'avoir les meilleures performances. L'algorithme des plus proches voisins est également très compétitif. Cependant, pour réaliser le système, un de ces deux algorithmes doit être implémenté. À cette fin, l'algorithme de *Random Forests* (forêts d'arbres décisionnels) a été ajouté au système, tout en gardant à l'esprit que ces deux algorithmes doivent être considérés dans l'ensemble des

algorithmes à tester. Chaque algorithme discuté à la section 2.2 sera testé. Les résultats obtenus seront présentés au chapitre 5 et viendront justifier le choix des forêts d'arbres décisionnels. Un second point à aborder est présenté : les exercices et AVQ qui sont reconnues, et comment elles sont reconnues. En effet, la problématique de ce mémoire demande une reconnaissance sur deux types d'activités différentes. Or, dans la littérature récente étudiée à la section 2.3.2, les exercices physiques ou activités spécifiques étaient en tout temps effectués avec le même algorithme. Par exemple dans les derniers travaux étudiés, l'équipe avait effectué la reconnaissance d'activités entre un coup de genou sur le même fenêtrage que l'activité de marcher, ce qui peut laisser perplexe. À la lumière de ce qui a pu être observé dans la littérature la plus récente possible, aucun travail ne permettait de répondre à la problématique. Le positionnement de ce mémoire dans le domaine de la recherche est alors sans équivoque.

## CHAPITRE 3

### PROTOTYPE

#### 3.1 INTRODUCTION

Le premier prototype est une preuve de concept. À cette fin, les dispositifs utilisés ont été choisis grâce à leur disponibilité, et leur utilisation rapide. Étant donné la problématique de ce mémoire, il fallait utiliser un périphérique de collecte de données accélérométriques afin de récupérer les données nécessaires à la reconnaissance d'activités. Ces données permettent une analyse rendant possible une classification des activités physiques effectuées. Dans la revue de littérature effectuée à la section 2.3, il a été remarqué que beaucoup de chercheurs utilisaient un tel dispositif placé au poignet. En fonction de la population ciblée, cet emplacement est le plus judicieux. En effet, pour les personnes de tous âges, le poignet est une zone du corps souvent sollicitée par des objets, comme des montres ou des bracelets divers. Afin de récolter les données accélérométriques, le choix du dispositif s'est porté sur un bracelet. Cependant, pour être en mesure d'analyser les données récoltées par ce dernier, un dispositif de communications sans fil apte à les recevoir en temps réel est nécessaire.

Beaucoup de périphériques pouvaient jouer le rôle du receveur des données émises par l'Intel Edison. L'exemple le plus évident aurait été un ordinateur. En effet, ce choix aurait permis d'avoir toutes les technologies possibles, afin d'effectuer un développement adaptatif si jamais la technologie choisie ne fonctionnait pas. Cependant, un ordinateur demande beaucoup de place. De plus, ces équipements sont souvent dotés de capteurs additionnels, comme une caméra, qui pourrait être extrêmement invasive pour un utilisateur. Afin de ne pas avoir ce genre de problème, la catégorie des nano-ordinateurs semble la plus cohérente. Le nano-ordinateur le plus performant pour cette tâche est le « Raspberry Pi 3 » (Raspberry Pi Foundation, 2016). Il sera étudié en détail dans la section 3.2.2.

## 3.2 MATÉRIEL

### 3.2.1 BRACELET – INTEL EDISON

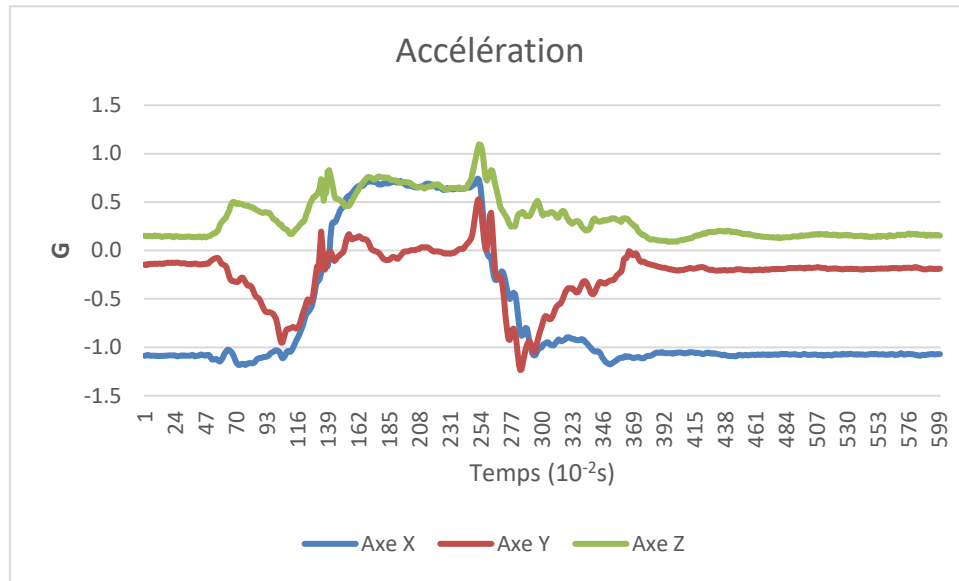
Le bracelet « *Intel Edison* », qui sera abrégé « Edison », est composé de plusieurs blocs (Intel, 2014). Chaque bloc possède une utilité précise, ce qui permet d'adapter la taille du prototype aux fonctionnalités désirées. Le bloc principal est l'Edison en lui-même : c'est un processeur Atom contenant deux cœurs cadencés à 500MHz, avec 4Gb de stockage Flash. Comme on peut le voir à la figure 8, ce bloc consiste en un simple circuit imprimé sur lequel le processeur est fixé.

Si un bracelet n'était constitué que de ce simple bloc, son utilité serait réduite à celle d'un nano-ordinateur basique. Cependant, beaucoup d'autres blocs existent et sont utilisés ici. Pour commencer, celui qui est le plus important est le bloc de batterie. Ce bloc permet d'alimenter l'ensemble du prototype final, peu importe s'il contient juste le processeur, ou un assemblage plus complexe. À des fins de prototypage et de développement, un autre bloc est également important, contenant le câblage et les soudures nécessaires pour une interaction directe avec le système d'exploitation de l'Edison. De plus, il est possible d'ajouter une carte SD dans une autre partie, afin de pouvoir stocker des données en plus grand nombre que sur la mémoire flash de l'Edison. Pour récolter les données pertinentes pour la reconnaissance d'activités physiques, il faut ajouter un autre bloc spécifique. Ce bloc contient une centrale inertielle à 9 degrés de liberté : il combine un accéléromètre, un gyroscope et un magnétomètre. De plus, l'accéléromètre, le gyroscope et le magnétomètre fournissent chacun les différentes variations sur les trois axes de l'espace, alors que certains ne permettent la



Figure 8 - Dispositif Intel Edison seul





**Figure 9 – Exemple d'accélération collectées lors de la réalisation d'une activité (squat sur chaise)**

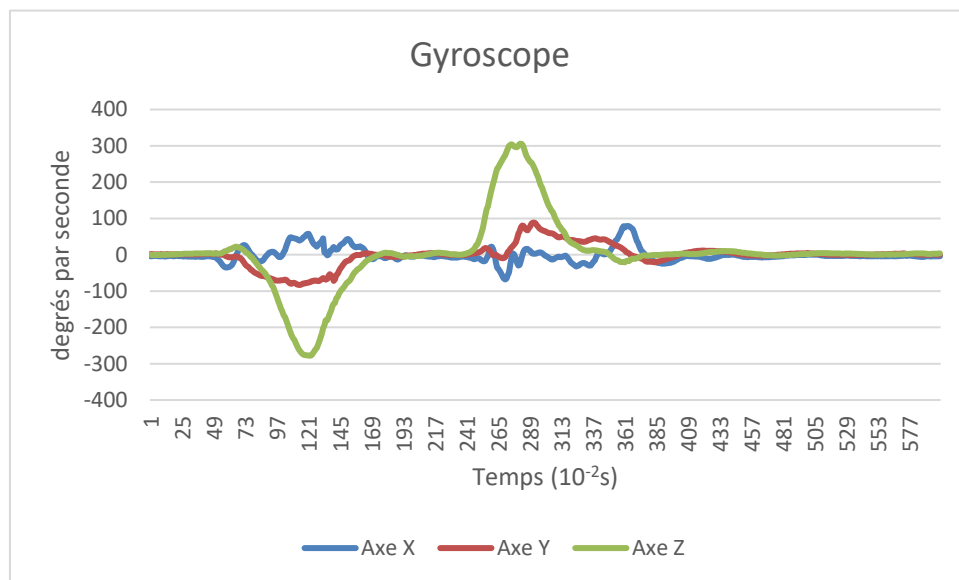
détection que sur deux. Afin de mieux comprendre leur utilité, une courte description de ce qu'ils permettent de récolter est donnée ci-après.

L'accéléromètre est un dispositif permettant de détecter et de mesurer les variations d'accélération. Il renvoie alors des mesures en G. Ce genre de périphérique est utilisé dans les fusées, les avions, et beaucoup d'autres véhicules ayant une accélération. Ces variations peuvent être analysées, comme le montre clairement la figure 9, lors de l'exécution d'un Squat (description de l'exercice dans le chapitre 4). On peut bien discerner qu'il y a des changements de valeurs conséquents lors de l'exécution d'activités quelconques, qui permettent de déterminer la présence d'une activité. Cependant, certaines activités ne peuvent être reconnues qu'à partir des données provenant de l'accéléromètre seul. C'est pourquoi le gyroscope est aussi mis à contribution.

Un gyroscope est un capteur servant à détecter et mesurer toutes les variations angulaires. Il va émettre des mesures en degrés par seconde. Il est beaucoup utilisé en aviation, ou même dans la station spatiale internationale, mais tend à se généraliser pour une

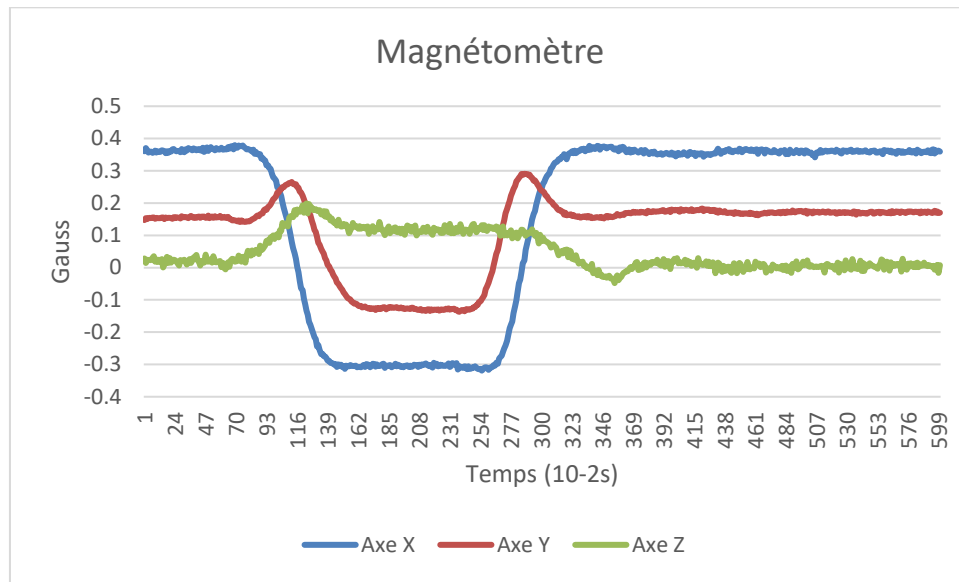
utilisation plus commune, comme dans des téléphones intelligents, ou des consoles de jeu (ex. : la console Wii de Nintendo). Dans le même esprit que pour l'accéléromètre, on peut facilement observer à la figure 10 ses variations pouvant aider à reconnaître les activités en cours.

Un dernier périphérique est contenu dans la centrale inertielle à 9 degrés de liberté : le magnétomètre. Le magnétomètre est un peu moins utilisé que les deux autres dispositifs présentés chez les constructeurs électroniques, mais est très utilisé dans la vie de tous les jours. Il permet de détecter les changements de champs magnétiques. Son unité est le Gauss. Il est généralement utilisé dans les sciences de la terre, afin de détecter des métaux ou des roches à résonance spécifique. Cependant, son utilisation s'est également généralisée, pour s'intégrer dans les téléphones intelligents, afin de créer des boussoles par exemple. Dans le domaine de la reconnaissance d'activités, on peut observer à la figure 11 qu'il peut fournir des données intéressantes.



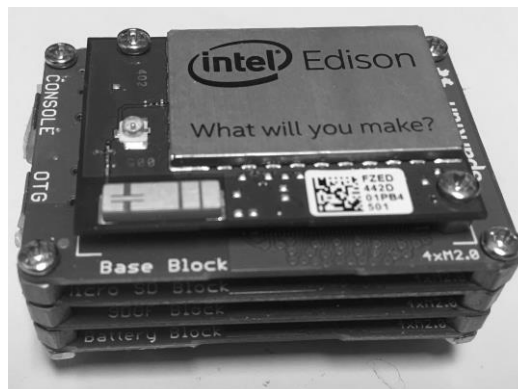
**Figure 10 – Exemple de variations collectées avec un gyroscope lors de la réalisation d'une activité (squat sur chaise)**

Dans le bloc de l'Edison, les trois capteurs définis ci-dessus sont réunis. Ils permettent alors de récupérer les données nécessaires à la reconnaissance d'activités physiques. Cependant, beaucoup d'autres recherches se basent sur l'utilisation d'un accéléromètre couplé avec un gyroscope. Le magnétomètre est présent dans beaucoup moins de travaux que ne le sont les deux autres capteurs discutés.



**Figure 12 – Exemple de variations collectées avec un magnétomètre lors de la réalisation d'une activité (squat sur chaise)**

Ainsi, le bracelet est composé de quatre blocs, et du processeur. Une image de l'Edison montée avec ses blocs est fournie dans la figure 12. De plus, l'ensemble des blocs composant ce bracelet est décrit de façon utilitaire dans le Tableau 2. Le prototype final est relativement petit, avec des dimensions d'environ 2.5cm de hauteur, 3cm de largeur, et 4.5cm de longueur.



**Figure 11 - Dispositif Intel Edison assemblé avec d'autres blocs**

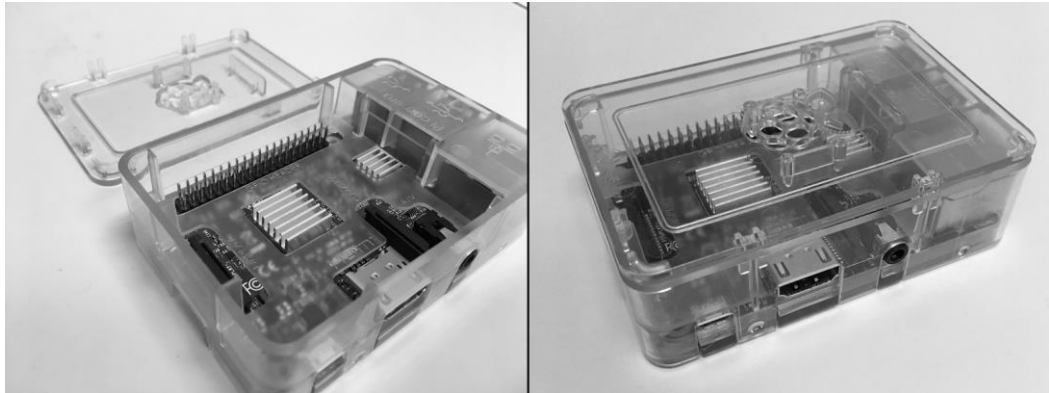
**Tableau 2 - Descriptif des blocs utilisés avec l'Edison**

« <b>Base Block</b> »	Ce bloc sert majoritairement au développement de programmes informatiques. On peut voir sur celui-ci (sur la gauche), deux textes indiquant « Console », et « OTG ». Ces deux sorties servent respectivement à interagir avec l'Edison directement, et à pouvoir accéder à ses fichiers dans la mémoire flash.
« <b>Micro SD Block</b> »	Comme son nom l'indique, ce bloc permet de stocker une carte SD, permettant ainsi de stocker des données accessibles par n'importe quel programme informatique. Son utilisation sera expliquée à la section 3.4 de ce chapitre.
« <b>9DOF Block</b> »	Ce bloc est celui contenant la centrale inertielle à 9 degrés de liberté. Ce bloc a donc le périphérique qui sert de capteur principal à notre système, constitué d'un accéléromètre, d'un gyroscope, et d'un magnétomètre.
« <b>Battery Block</b> »	Ce bloc de batterie contient une batterie lithium-ion polymère (LiPo) de 400 mAh. Cette batterie permet au prototype de continuer son exploitation sans être branché directement à une prise de courant.

Comme il a été énoncé plus tôt, l'Edison seul n'est pas capable d'effectuer simultanément la collecte de données, et de les analyser. Évidemment, il s'occupera de la collecte des données puisqu'il est muni des capteurs essentiels et peut aussi communiquer sans fil. Le périphérique de calcul et d'analyse sera alors confié au nano-ordinateur Raspberry Pi 3.

### **3.2.2 RASPBERRY PI 3**

Le nano-ordinateur *Raspberry Pi 3* est un des plus performants dans sa catégorie. Les récents travaux issus du domaine de la recherche l'utilisant prouvent bien son succès (Abrahamsson *et al.*, 2013; Tso *et al.*, 2013; Ferdoush, Li, 2014; Vujovic, Maksimovic, 2015). Afin d'alléger le texte, il sera abrégé par Raspberry seulement. La fondation l'ayant créé avait conçu plusieurs autres versions par le passé, mais la version 3 se démarque grandement de ses prédécesseurs. En effet, celle-ci apporte une évolution importante du processeur, le munissant de quatre cœurs 64 bits cadencés à 1.2GHz. De plus, l'intégration des technologies sans fil Wi-Fi et Bluetooth l'a propulsé au sommet de la popularité. Une image de celui-ci dans sa coque est illustrée à la figure 13.



**Figure 13 - Raspberry Pi 3**

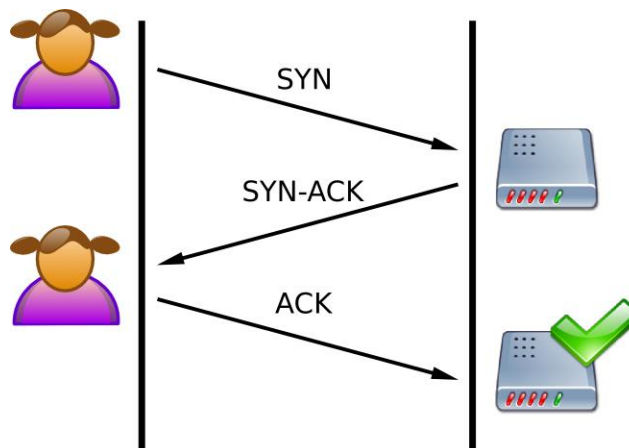
Dans ce mémoire, il est utilisé pour recevoir les données collectées par l'Edison, et appliquer des algorithmes d'apprentissage et de classification. En effet, comme mentionné précédemment, l'Edison n'étant pas assez performant pour réaliser ces deux tâches, il a fallu déléguer la seconde au Raspberry qui est en mesure d'effectuer d'importantes quantités de calculs chaque seconde. De plus, l'exécution d'algorithmes du domaine du forage de données nécessite plus de puissance que la tâche de collecte de données, et donc plus d'énergie. De ce fait, le Raspberry est le candidat idéal, étant donné que ce dispositif est branché directement sur le secteur, ce qui lui permet de n'avoir aucune contrainte énergétique. Par ailleurs, ce nano-ordinateur possède une multitude de protocoles de communications sans fil. Le principal défi dans la réalisation de ce prototype consistait alors à développer le module permettant la communication entre le Raspberry et l'Edison. Étant donné les protocoles disponibles sur chaque périphérique, il a fallu expérimenter plusieurs protocoles afin de choisir le plus pertinent et le plus performant.

### **3.3 COMMUNICATION**

Plusieurs moyens de communication étaient disponibles sur chacun des périphériques utilisés. Cependant, à des fins de prototypage, la communication par le biais du Wi-Fi est choisie, étant donné la facilité d'utilisation de celui-ci par rapport aux autres. Or, avec ce mode

de communication, il existe deux protocoles : le TCP (*Transmission Control Protocol*) et l'UDP (*User Datagram Protocol*). Ces deux protocoles sont expliqués et comparés en détail ci-après.

Tout d'abord, le protocole TCP favorise la qualité des communications plutôt que leur quantité. Il permet d'assurer que chaque périphérique relié par ce protocole connaît l'état de la communication courante. Pour ce faire, lorsqu'un des dispositifs envoie des données à l'autre, il lui envoie le message « SYN », et va attendre que son partenaire lui confirme la réception avant de passer à autre chose. Cette réponse est exprimée par le message « SYN-ACK ». Elle provient du receveur qui demande une synchronisation d'état, lors de la réception d'un paquet de données. À ce moment, l'émetteur des données renvoie alors le signal « ACK », qui signifie qu'il a bien compris que le receveur a reçu les données. Une fois cet échange de signaux terminé, les deux périphériques peuvent alors se transmettre de nouveaux paquets de données. Le fonctionnement est illustré à la figure 14.



**Figure 14 – Fonctionnement du protocole TCP**

Le protocole TCP est très utilisé pour la sécurité du transfert de données qu'il permet. Par exemple, il intervient lors de la connexion à un service internet ou à tout ce qui concerne un échange de fichiers. De façon concrète, un utilisateur envoyant une photo à un de ses amis

aime savoir que sa photo est arrivée dans son intégralité. Ce genre de confirmation est assurée par défaut grâce au protocole TCP.

Le protocole UDP quant à lui n'a besoin d'aucune connexion persistante entre deux utilisateurs. Dans ce cadre, aucun signal tel que le « SYN » ou le « ACK » n'est attendu. Par définition, il n'assure donc pas que chaque paquet de données a été reçu par un des partis. Cependant, il permet d'avoir un débit beaucoup plus élevé, dû au fait que la vérification de réception d'un paquet de données est inexistante. Un cas d'utilisation typique de ce protocole est celui des appels vidéo, ou de tout flux de média continu. En effet, il est préférable lors d'un appel vidéo, de rater quelques images avec un taux continu de 50 images par secondes grâce au protocole UDP, plutôt que d'avoir à coup sûr 10 images par secondes grâce au protocole TCP (10 images par seconde est très saccadé, et par conséquent vite agaçant pour l'utilisateur). La différence de débit entre ces deux protocoles est alors impressionnante.

Étant donné que chaque protocole de communication observé (TCP et UDP) possède ses forces et ses faiblesses, il a fallu les tester tous les deux. Pour se faire, un réseau Wi-Fi avec un routeur performant a été réservé pour effectuer l'expérimentation, afin d'avoir la certitude de n'avoir aucune perte due au réseau. Les deux protocoles ont alors été testés à leur pleine puissance, sans aucune bride logicielle, pendant une minute continue. Les données envoyées lors du test sont réelles, et représentent des données accélérométriques récoltées par l'Edison. Le protocole TCP a été testé en premier. Le protocole a alors résulté en un flux continu de données, à un taux d'envoi d'environ 20Hz. C'est un taux qui est très correct pour effectuer de la reconnaissance d'AVQ, comme dans les travaux de (Bao, Intille, 2004), mais trop faibles pour identifier correctement des exercices physiques plus complexes. Ensuite, le protocole UDP a été testé avec les mêmes données. Par rapport au protocole TCP, le protocole UDP a permis d'obtenir un taux d'envoi d'environ 100 Hz, ce qui est très impressionnant. Cependant, UDP a également une réputation comme quoi les paquets qu'il envoie peuvent arriver dans un ordre différent de l'ordre d'envoi. Ceci pouvant perturber les

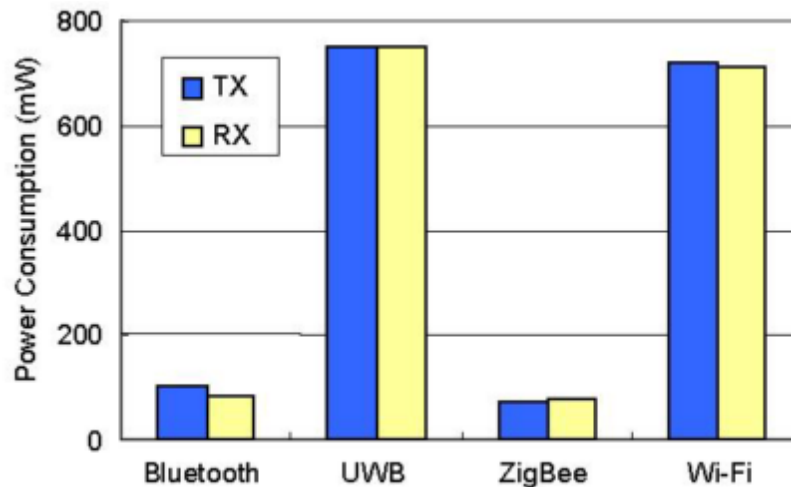
données, elles ont été examinées à la sortie de l'Edison, et à l'arrivée du Raspberry. Sur la minute de données, 6 000 enregistrements ont été produits, contenant eux-mêmes 9 axes de données (dus à la centrale inertielle à 9 degrés de liberté). Sur ces 54 000 données, il n'y a eu aucune perte et aucun changement d'ordre. Aucune perte n'ayant été occasionnée, et aucun désordre dans la succession des données, le protocole UDP a alors été favorisé par rapport au protocole TCP dans ce mémoire. Au vu de ce test, même si une perte de données était envisageable, la perte serait tellement minime qu'elle n'influerait que trop peu sur l'algorithme de reconnaissance pour ne pas prendre en compte ce protocole.

En revanche, un cas problématique était envisageable lors d'une utilisation réelle. La technologie Wi-Fi est à portée relativement courte (plusieurs dizaines de mètres), que se passerait-il si le patient portant l'Edison se retrouvait hors de portée du Raspberry ? La solution utilisée dans ce mémoire est de transférer les données non transmissibles vers un stockage annexe, facilement récupérable. À la Figure 11, on pouvait voir un bloc comportant une carte SD : c'est cette carte qui stocke alors les données qui ne peuvent pas être envoyées au Raspberry. Une carte SD de 4Go pourrait alors contenir environ 115 millions d'enregistrements, ce qui représenterait 319 heures de données continues, ce qui est largement suffisant, étant donné que la batterie ne tiendrait pas autant de temps.

À ce sujet, ce premier prototype possède un point faible important : son autonomie. En effet, si la technologie Wi-Fi est très pratique pour réaliser des preuves de concept, elle n'est pas très performante en termes d'énergie en cas réel. En effet, le temps d'utilisation du prototype est considérablement réduit à cause de ce moyen de communication (environ 2h30 d'autonomie avant que la batterie ne soit complètement déchargée). Une solution serait alors de changer ce mode de communication entre l'Edison et le Raspberry. De plus, ce changement permettrait de soustraire au système l'utilisation d'un routeur, et donc de nécessiter encore moins de place. Une étude de (Lee *et al.*, 2007) porte sur les différences entre les protocoles sans fils de nos jours. Une technologie moins énergivore se révélerait être le Bluetooth. En



outre, la figure 15 issue de cet article exprime bien la différence de consommation d'énergie entre le Wi-Fi et le Bluetooth. Il est à noter que TX représente la consommation d'énergie lors de l'envoi de données, tandis que RX représente la consommation d'énergie lors de la réception de données.



**Figure 15 - Consommation d'énergie des différents protocoles (Lee et al., 2007), p6**

Comme on peut le voir à la figure 15, le Wi-Fi consomme environ 5 fois plus d'énergie que le Bluetooth. Cependant, l'article cité a été écrit en 2007. Depuis cette date, le Bluetooth a connu beaucoup d'améliorations et notamment en 2010, lorsque le Bluetooth à basse consommation (*Bluetooth Low Energy : BLE*) est apparu. Comme son nom l'indique, cette version a une consommation énergétique bien inférieure à son prédécesseur. De plus, il permet un débit théorique du même ordre de grandeur, à savoir environ 1Mb/s, ce qui représenterait plus de 3000Hz de débit, ce qui est plus que suffisant. Ce serait alors le mode de communication idéal pour remplacer le Wi-Fi, et rendre le prototype plus performant en conditions réelles. Cependant, cette technologie n'est pas stable sur l'Edison, rendant son évolution avec le matériel actuel impossible. Pour faire évoluer le projet, il faudrait créer une nouvelle version du bracelet.

### 3.4 VERSION 2.0 DU BRACELET

Le principal défaut du premier prototype était son autonomie. On a vu dans la section précédente que la technologie Wi-Fi était la perte principale d'énergie. Afin d'optimiser la gestion d'énergie de ce nouveau prototype, il faut changer le moyen de communication. Le BLE semble alors être la meilleure option. De plus, le précédent bracelet possédait une batterie de 400 mAh, ce qui est une petite batterie. Dépendamment de la taille finale du prototype, celle-ci pourrait éventuellement être changée pour une batterie Lithium-Ion Polymer (*LiPo*) de 800 mAh qui n'est pas beaucoup plus volumineuse.

Afin de réaliser ce nouveau bracelet, il a fallu trouver un périphérique capable d'allier le fait d'être petit, d'avoir une connexion stable via le BLE et de pouvoir collecter les données en continu sans ralentir. Plusieurs dispositifs ont été testés, notamment le Flora de la gamme d'Adafruit (Adafruit, 2015), mais les essais n'ont pas été fructueux en termes de performance. Cependant, la puce de *Nordic Semi* « *NRF52832* » (NordicSemi, 2016) s'est révélée très prometteuse (celle-ci sera abrégée NRF dans ce mémoire). On peut observer la carte de développement de cette puce à la figure 16, mais la taille de la puce elle-même représente uniquement le petit carré blanc que l'on peut y voir. Elle possède un système embarqué permettant d'être autonome en termes de connectique. En effet, le Flora avait besoin d'une puce pour gérer un programme mère, tandis que le NRF gère lui-même ce genre de détails.



**Figure 16 - Carte de développement du NRF52832**

De plus, ce périphérique a été conçu initialement pour communiquer via le protocole BLE, ce qui le rend candidat naturel pour remplacer l'Edison. Cependant, cette puce ne permet que d'effectuer les calculs et de communiquer, elle ne permet pas de récolter de données accélérométriques à elle seule. Il est en revanche possible d'ajouter une centrale inertielle à 9 degrés de liberté sur certaines des nombreuses sorties qu'elle possède (SPI, I<sup>2</sup>C, UART). Par ailleurs, il existe beaucoup de centrales inertielles différentes, mais étant donné que celle de l'Edison fonctionnait bien, et qu'une partie de l'adaptation était déjà développée, il est plus judicieux d'utiliser une centrale de la même gamme. Celle de l'Edison était alors un LSM9DS0 (ST-Microelectronics, 2013a), mais pour des raisons pratiques, le LSM9DS1 (ST-Microelectronics, 2013b) est plus pertinent. La différence entre ces deux centrales inertielles se situe surtout au niveau des différents paramétrages disponibles de chaque capteur, ce qui n'influence pas leur rendement à l'échelle de ce mémoire. Afin de recréer le même type de bracelet qu'avec l'Edison, un autre composant doit être ajouté : la carte SD. Cependant, le prototype n'en contient pas, mais un tel composant serait relativement simple à intégrer étant donné la quantité de sorties disponibles sur le NRF. Une fois le matériel prototypé, il faut penser à l'élaboration de la solution logicielle de celui-ci.

Le BLE fonctionne par services. Chaque service doit idéalement représenter une partie de données. De plus, chaque service contient une liste de caractéristiques représentant son contenu. Par exemple un service nommé « Accéléromètre » contiendrait 3 caractéristiques « Axe X », « Axe Y », et « Axe Z ». Ceci permettrait une meilleure lisibilité des données sur le périphérique BLE. Cependant, actualiser 9 caractéristiques à la fois, et les émettre est une tâche demandant une certaine puissance, et donc une certaine énergie. Une façon d'optimiser ce rendement, et de rendre la communication plus fluide est de combiner toutes les valeurs de chaque axe en une seule valeur, pour la transmettre par le biais d'une seule caractéristique. Dans ce cas, les transferts de données sont 9 fois moins importants, étant donné qu'il n'y a qu'un seul envoi de données, et ce, même si le paquet envoyé est plus conséquent. Cette façon de faire est alors la plus représentative d'une optimisation d'autonomie, sans perdre de

performance. La version du bracelet élaborée dans cette section est en cours de développement, et sera terminée sous peu. La complétion de ce second prototype dépassait largement le cadre de ce mémoire.

## **CHAPITRE 4**

### **MODÈLE DE RECONNAISSANCE HYBRIDE**

#### **4.1 INTRODUCTION**

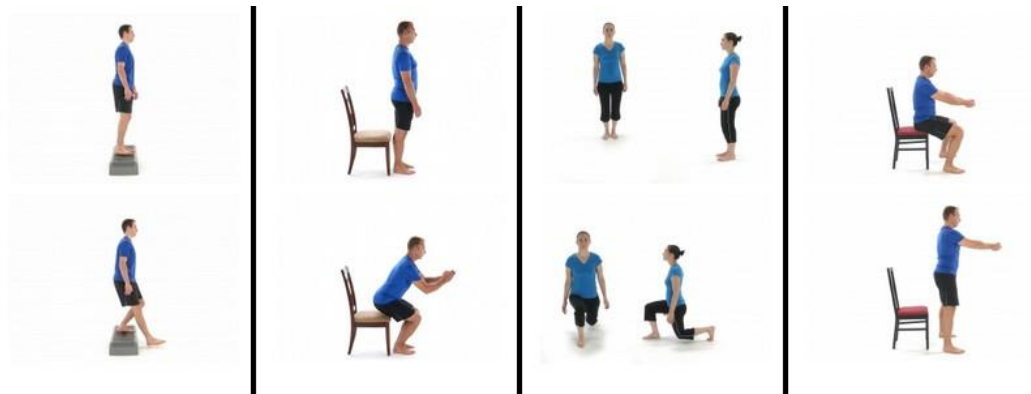
Dans la littérature, il a été observé qu'aucun travail à ce jour ne portait sur la reconnaissance d'AVQ et d'exercices physiques sur deux fenêtres de temps différentes simultanément. Dans le cadre de ce mémoire, des activités de la vie quotidienne et des exercices physiques prescrits par un thérapeute doivent pouvoir être reconnus avec un seul et même algorithme, malgré leurs différences de fenêtrage. Dans ce chapitre, les différentes AVQ et les exercices physiques seront décrits (tels qu'ils pourraient être prescrits en conditions réelles). De plus, afin de reconnaître des activités à l'aide de données accélérométriques, il est nécessaire de calculer des attributs (comme certains ont été énoncés à la section 2.3). En s'inspirant des attributs efficaces de la littérature, un total de 35 attributs ont été extraits des données de chaque capteur (accéléromètre, gyroscope, magnétomètre). Ils seront alors tous décrits à la section 4.3 suivante. Pour finir, la méthode de reconnaissance hybride conçue dans ce mémoire sera alors expliquée en détail.

#### **4.2 PRÉSENTATION DES AVQ ET DES EXERCICES PHYSIQUES**

Afin de quantifier le niveau d'activité global d'une personne, il est possible d'utiliser les AVQ. On peut observer une grande majorité de celles-ci dans les différents travaux mentionnés à la section 2.3. Les activités les plus communes, qui sont également celles qui seront utilisées dans ce mémoire, sont listées ci-après : « marcher », « courir », « se lever », « s'asseoir », « rester assis ». Cependant, afin d'indiquer au système que l'utilisateur va réaliser ses exercices prescrits, il est nécessaire d'ajouter une sixième activité, qui consiste en une rotation rapide du poignet. Son utilisation concrète sera expliquée plus tard dans ce chapitre. Au total, on retrouve alors six AVQ pour quantifier le niveau d'activité global d'une personne. Par ailleurs, ces activités sont relativement rapides à effectuer. Après une analyse de la littérature,

et quelques essais, la fenêtre d'exécution de ces AVQ a été fixée à deux secondes, ce qui est très différent des exercices physiques étudiés.

Sur ce point, désirant avoir une étude la plus proche de la réalité possible, des exercices physiques ont été fournis tels qu'un thérapeute pourrait les prescrire pour un patient atteint de DM1. Ce sont alors de véritables exercices, nécessitant plus de précision, et donc plus de données que les AVQ énoncées plus tôt. Afin d'augmenter la quantité de données récoltées, il faut augmenter le fenêtrage ; après plusieurs essais, la fenêtre d'exécution des exercices physiques a été fixée à six secondes. Ce long temps est surtout dû au fait qu'une pause de deux secondes doit être respectée pendant la réalisation de chaque exercice, afin que les muscles travaillent. Il a alors été observé qu'il fallait environ deux secondes pour accéder à la pose de l'exercice, et deux secondes supplémentaires pour revenir à la position initiale, totalisant alors les six secondes proposées. À la figure 17, les quatre exercices physiques considérés sont présentés. Une description de chacun d'eux est donnée dans le Tableau 3, dans le même ordre d'apparition qu'à la figure 17. La description contient également le détail de la réalisation de l'exercice concerné.



**Figure 17 - Exercices physiques étudiés**

**Tableau 3 - Description des exercices physiques étudiés**

Stabilisation descente marche	<ul style="list-style-type: none"> <li>- Se placer debout sur une marche, le dos bien droit.</li> <li>- Décoller le pied et l'avancer devant en pliant le genou opposé, comme pour descendre de la marche, mais sans toucher le sol. Rester deux secondes dans cette position, en s'assurant que la rotule de la jambe d'appui soit alignée avec le deuxième orteil du pied.</li> <li>- Retourner à la position initiale en remettant le pied sur le tabouret.</li> <li>- Répéter autant de fois qu'énoncé par le thérapeute, en échangeant le pied à chaque fois.</li> </ul>
Squat sur chaise	<ul style="list-style-type: none"> <li>- Débuter debout devant une chaise.</li> <li>- Pousser les hanches vers l'arrière comme pour aller s'asseoir sur la chaise, en pliant les genoux afin de toucher la chaise avec les fesses.</li> <li>- Tenir deux secondes cette position.</li> <li>- Revenir ensuite à la position debout.</li> <li>- Il faut toujours garder les genoux alignés avec le deuxième orteil de chaque pied.</li> <li>- Répéter autant de fois que demandé par le thérapeute.</li> </ul>
Fente avant alternée	<ul style="list-style-type: none"> <li>- Se tenir debout, les genoux à la largeur des hanches.</li> <li>- Faire un grand pas devant, et plier le genou pour tenter d'obtenir un angle de 90° sur chaque genou.</li> <li>- Garder le tronc et le bassin le plus stable possible, et le pied aligné avec le genou et la hanche.</li> <li>- Rester deux secondes dans cette position.</li> <li>- Revenir dans la position initiale</li> <li>- Répéter autant de fois que nécessaire, en alternant la jambe.</li> </ul>
Assis à debout	<ul style="list-style-type: none"> <li>- S'asseoir sur une chaise placée contre un mur.</li> <li>- Croiser les doigts et étirer les bras devant.</li> <li>- Avec les pieds à la largeur des hanches, et les fesses au bout de la chaise, soulever les hanches afin de se lever.</li> <li>- Tenir deux secondes au maximum.</li> </ul>

	- Revenir en position assise, et répéter autant de fois que nécessaire.
--	---

### 4.3 ATTRIBUTS UTILISÉS

Pour la reconnaissance d'activités à l'aide de données accélérométriques, il est important d'utiliser des algorithmes du domaine du forage de données. En effet, d'autres méthodes pourraient être envisagées, mais les algorithmes d'apprentissage se révèlent très souvent plus efficaces. Cependant, ce type d'algorithme a besoin de données formatées de manière à ce qu'une classe soit définie par plusieurs attributs. Un exemple simple de ceci serait, par exemple, pour la reconnaissance de types de fleurs où les attributs seraient : la longueur de ses pétales, la largeur de celles-ci, leur couleur, etc. Les algorithmes pourront alors analyser ces attributs afin d'identifier la classe d'une éventuelle nouvelle fleur inconnue. En revanche, dans le cadre de ce mémoire, les activités à détecter se produisent sur plusieurs secondes. Il faut alors regrouper ces multiples enregistrements en un seul avec des attributs : ils seront alors la représentation d'une itération de l'activité réalisée. Lors de la revue de littérature, plusieurs attributs ont été mentionnés et utilisés dans différents travaux. Parmi l'ensemble des attributs pouvant être utilisés, on peut distinguer deux grands types : ceux appartenant au domaine temporel et ceux appartenant au domaine fréquentiel. Ces attributs seront alors abordés et explicités en fonction de leur type, en commençant par ceux du domaine temporel.

Chaque attribut présent dans le domaine temporel est basé sur le temps. De ce fait, comme la quantité de données récoltées est proportionnelle au temps (étant donné la fréquence fixe), chaque attribut est alors calculé en fonction du nombre total d'enregistrements dans la fenêtre qui le concerne. De plus, un attribut est calculé en fonction d'un ou de plusieurs axes d'un seul capteur. Par exemple, certains peuvent utiliser un seul axe de l'accéléromètre, ou d'autres peuvent utiliser tous les axes de ce même accéléromètre. Cependant, étant donné la quantité gigantesque d'attributs envisageables dans le domaine temporel, un choix a dû être



réalisé en fonction de leur pertinence. Dans les travaux de (Cleland *et al.*, 2013), on peut observer beaucoup d'attributs différents de ce domaine. Tout d'abord, la moyenne des valeurs sur chaque axe est calculée. C'est un attribut très basique, mais qui permet d'avoir un bon aperçu des données recueillies sur la fenêtre de temps. La moyenne de ces moyennes est également calculée comme un deuxième attribut. En liaison avec ces attributs, l'écart-type est aussi sélectionné, et il en va de même pour l'écart-type moyen, représentant la moyenne de ce même attribut en combinant les trois axes. Les deux prochains attributs sont moins connus, mais restent très utilisés dans le domaine de la reconnaissance d'activités : le coefficient d'asymétrie (*skewness*) d'une distribution statistique et celui d'aplatissement (*kurtosis*). Ils sont utilisés en particulier dans les travaux de (Jolliffe, 2002; Krishnan, Cook, 2014). Les définitions mathématiques de l'ensemble des attributs énoncés sont données au Tableau 4. La moyenne respective des coefficients d'asymétrie et des coefficients d'aplatissement est également ajoutée. Dans les travaux de (Gao *et al.*, 2014), ils utilisent un autre attribut simple qui permet d'avoir de bons résultats : le *Zero-Crossing Rate* (ZCR). C'est le taux de passage par zéro, à savoir le nombre de fois que les données repassent par la valeur 0 pendant l'enregistrement. Cet attribut est calculé pour chaque axe, et la moyenne des ZCR est ajoutée également. Le dernier attribut temporel est celui des corrélations. C'est un attribut calculé entre chaque paire d'axes, comme le ratio entre la covariance entre ces deux axes, et le produit de leurs écarts-types respectifs. Le tableau suivant résume chaque attribut énoncé et leur définition mathématique.

**Tableau 4 – Définitions mathématiques des attributs du domaine temporel**

<p>Afin de simplifier les équations suivantes, les variables suivantes doivent être définies :</p> <ul style="list-style-type: none"> <li>- « <math>x_i</math> » correspond à une valeur où :  <math>x</math> correspond à l'axe et <math>i</math> correspond au numéro de la valeur dans l'ensemble.</li> <li>- <math>E</math> est l'ensemble contenant l'ensemble des <math>x_i</math> avec <math>i</math> allant de 1 à <math>N</math> (<math>N</math> étant le nombre de valeurs dans l'ensemble <math>E</math>)</li> </ul>	
Moyenne pour un axe donné	La valeur $\bar{x}$ dénote la moyenne sur l'axe $x$ et se calcule de la manière suivante :

	$\bar{x} = \frac{\sum_{i=1}^N x_i}{N}$ <p><b>Équation 14 - Calcul de la moyenne pour un axe donné</b></p>
Moyenne des moyennes de chaque axe	<p>Soient <math>\bar{x}</math>, <math>\bar{y}</math>, <math>\bar{z}</math> les moyennes respectives des trois axes du capteur. La moyenne des moyennes de chaque axe s'obtient de la manière suivante :</p> $\overline{XYZ} = \frac{\bar{x} + \bar{y} + \bar{z}}{3}$ <p><b>Équation 15 - Calcul de la moyenne des moyennes de chaque axe</b></p>
Écart-type pour un axe donné	<p>La valeur <math>\sigma_x</math> représente la valeur de l'écart-type sur l'axe <math>x</math>, et peut être obtenue avec la formule suivante :</p> $\sigma_x = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$ <p><b>Équation 16 - Calcul de l'écart-type pour un axe donné</b></p>
Moyenne des écart-types des axes	<p>Soient <math>\sigma_x</math>, <math>\sigma_y</math>, <math>\sigma_z</math> les valeurs de l'écart-type obtenues pour chacun des trois axes. La moyenne des écart-types s'obtient par la formule suivante :</p> $\bar{\sigma} = \frac{\sigma_x + \sigma_y + \sigma_z}{3}$ <p><b>Équation 17 - Calcul de la moyenne des écart-types de chaque axe</b></p>
Skewness pour un axe donné	<p>Soit <math>Skew_x</math> la valeur de coefficients d'asymétrie pour l'axe <math>x</math>. Cette dernière se calcule à l'aide de la formule qui suit :</p> $Skew_x = \frac{N}{(N-1)(N-2)} \sum_{i=1}^N \frac{(x_i - \bar{x})^3}{\sigma_x^3}$ <p><b>Équation 18 - Calcul de Skewness pour un axe donné</b></p>
Moyenne des Skewness pour chaque axe	<p>Soient <math>Skew_x</math>, <math>Skew_y</math>, <math>Skew_z</math> les valeurs de coefficients d'asymétrie obtenues pour chacun des trois axes. Leur moyenne s'obtient à l'aide de la formule suivante :</p> $\overline{Skew} = \frac{Skew_x + Skew_y + Skew_z}{3}$ <p><b>Équation 19 - Calcul de la moyenne des Skewness pour chaque axe</b></p>
Kurtosis pour un axe donné	<p>Soit <math>Kurt_x</math> la valeur de coefficients d'aplatissement pour l'axe <math>x</math>. Cette dernière se calcule à l'aide de la formule qui suit :</p>

	$Kurt_x = \frac{N(N+1)}{(N-1)(N-2)(N-3)} \sum_{i=1}^N \left( \frac{(x_i - \bar{x})^4}{\sigma_x^4} - \frac{3(N-1)^2}{(N-2)(N-3)} \right)$ <p><b>Équation 20 - Calcul de Kurtosis pour un axe donné</b></p>
Moyenne des Kurtosis pour chaque axe	<p>Soient <math>Kurt_x</math>, <math>Kurt_y</math>, <math>Kurt_z</math> les valeurs de coefficients d'aplatissement obtenues pour chacun des trois axes. On obtient leur moyenne par la formule suivante :</p> $\overline{Kurt} = \frac{Kurt_x + Kurt_y + Kurt_z}{3}$ <p><b>Équation 21 - Calcul de la moyenne des Kurtosis pour chaque axe</b></p>
ZCR sur un axe donné	<p>Soit <math>ZCR_x</math> la valeur du taux de passage par le zéro sur l'axe <math>x</math>.</p> <p>Soit également la fonction <math>\mu</math> qui retourne 1 si la condition énoncée est vraie, et 0 dans le cas contraire.</p> <p>La formule pour obtenir ce ZCR est alors :</p> $ZCR_x = \frac{1}{N} \sum_{i=2}^N \mu\{x_i x_{i-1} < 0\}$ <p><b>Équation 22 - Calcul du ZCR sur un axe donné</b></p>
Moyenne des ZCR pour chaque axe	<p>Soient <math>ZCR_x</math>, <math>ZCR_y</math>, <math>ZCR_z</math> les valeurs du taux de passage par le zéro obtenues pour chacun des trois axes. On obtient leur moyenne par la formule suivante :</p> $\overline{ZCR} = \frac{ZCR_x + ZCR_y + ZCR_z}{3}$ <p><b>Équation 23 - Calcul de la moyenne des ZCR pour chaque axe</b></p>
Corrélation entre deux axes	<p>Soit <math>Corr_{x,y}</math> la corrélation entre l'axe <math>x</math> et l'axe <math>y</math>.</p> <p>Dans le cadre de ce mémoire, la corrélation est effectuée entre chaque paire d'axe, et le total de chaque axe. Il y a donc 6 corrélations, résumées par les paires avec la liste suivante :</p> <ul style="list-style-type: none"> <li>- Axe x et axe y</li> <li>- Axe x et axe z</li> <li>- Axe x et total</li> <li>- Axe y et axe z</li> <li>- Axe y et total</li> <li>- Axe z et total</li> </ul> <p>L'équation permettant d'avoir la corrélation entre les deux axes concernés est la suivante (ici avec l'axe <math>x</math> et l'axe <math>y</math>) :</p> $Corr_{x,y} = \frac{cov_{x,y}}{\sigma_x \sigma_y}$ <p><b>Équation 24 - Calcul de la corrélation entre deux axes</b></p>

	<p>où <math>cov_{x,y}</math> est la covariance entre deux axes définie par l'équation suivante :</p> $cov_{x,y} = \frac{\sum_{i=1}^N ((x_i - \bar{x})(y_i - \bar{y}))}{N - 1}$ <p><b>Équation 25 - Calcul de la covariance entre deux axes</b></p>
--	--

Dans le cadre de ce mémoire, une centrale inertielle à 9 degrés de liberté est utilisée. Cette centrale est composée de trois capteurs, ayant chacun trois axes. En suivant correctement l'ensemble des attributs énoncés ci-avant, on peut observer qu'il y a 26 attributs du domaine temporel pour chaque capteur, ce qui occasionne au total 78 attributs du domaine temporel.

Néanmoins, malgré que l'utilisation d'attributs d'un seul domaine soit efficace, l'utilisation couplée d'attributs du domaine temporel et fréquentiel augmente le taux de reconnaissance (Mannini *et al.*, 2013). Ce domaine est caractérisé par l'utilisation de la transformée de Fourier rapide (FFT) (Brigham, Morrow, 1967). Ce calcul permet de transformer des données issues du domaine temporel en des données du domaine fréquentiel. Ces données sont très souvent des nombres complexes (i.e. elles ont une partie dite réelle et une autre dite imaginaire). À partir de ces nouvelles données, on peut alors calculer de nouveaux attributs qui seront du domaine fréquentiel. De la même manière que pour les attributs temporels, chaque attribut fréquentiel est détaillé dans le Tableau 5. Pour commencer, dans les travaux de Cleland *et al.*, (2013), un attribut fréquentiel est utilisé : l'énergie. C'est un attribut très connu pour sa diversité de valeurs ; en effet, des activités comme « rester assis » ne donnent pas de variation d'énergie conséquente, tandis que l'énergie calculée sur une activité comme « courir » est plus importante. Dans les travaux de (Bao, Intille, 2004), un autre attribut fréquentiel est utilisé : l'entropie. Pourtant, malgré qu'il soit différent, le calcul de l'entropie nécessite au préalable d'avoir les valeurs de l'énergie. Un troisième attribut du domaine fréquentiel est utilisé dans ce mémoire : le coefficient DC. Il représente la moyenne de la norme fréquentielle : elle est calculée à partir de la partie réelle des nombres complexes.

**Tableau 5 - Définitions mathématiques des attributs du domaine fréquentiel**

<p>Pour rendre plus simples les équations suivantes, plusieurs variables sont définies.</p> <p>Soit alors l'ensemble <math>E</math> représentant les valeurs résultantes de l'application de la FFT pour l'ensemble de données concerné, et <math>N</math> sa taille.</p> <p>Soient également pour chaque nombre complexe compris dans <math>E</math> deux variables <math>a_x</math> et <math>b_x</math> représentant respectivement la partie réelle et la partie imaginaire d'un nombre complexe.</p>	
Énergie	<p>La formule à appliquer pour calculer l'énergie est définie par :</p> $Energie = \frac{1}{N} \sum_{i=1}^N (a_i^2 + b_i^2)$ <p><b>Équation 26 - Calcul de l'énergie</b></p>
Entropie	<p>La formule pour calculer l'entropie est la suivante :</p> $Entropie = - \sum_{i=1}^N (P_i \ln P_i)$ <p><b>Équation 27 - Calcul de l'entropie</b></p> <p>Où la variable <math>P_i</math> est définie par l'équation suivante :</p> $P_i = \frac{a_i^2 + b_i^2}{N - Energie}$ <p><b>Équation 28 - Sous-calcul nécessaire pour le calcul de l'énergie</b></p>
Coefficient DC	<p>Le calcul à appliquer pour avoir le coefficient DC est le suivant :</p> $DC = \frac{1}{N} \sum_{i=1}^N a_i^2$ <p><b>Équation 29 - Calcul du coefficient DC</b></p>

Comme mentionné précédemment, la centrale inertielle utilisée possédant 9 axes, et le nombre d'attributs fréquents s'élevant au nombre de 3, il y a alors 27 attributs fréquents au total. En ajoutant à ce nombre le nombre d'attributs temporels préalablement détaillés, le nombre final d'attributs utilisés dans ce mémoire est de 105. C'est un nombre imposant, c'est pourquoi la puissance de calcul apportée par le Raspberry est importante : autant de calculs n'auraient jamais pu être envisageables sur l'Edison seul.

Lorsque l'ensemble de tous ces attributs est calculé, le jeu de données est alors sous une forme de plusieurs attributs pour une classe, qui est le format adéquat à l'utilisation d'algorithmes d'apprentissage observés dans le chapitre 2. Une utilisation spéciale est cependant requise pour répondre correctement à la problématique posée. En effet, les algorithmes d'apprentissages ne peuvent pas reconnaître deux activités sur des fenêtrages différents, il faut alors adapter la solution à ce cas. La solution est abordée à la section suivante.

#### **4.4 MÉTHODE DE RECONNAISSANCE HYBRIDE**

Répondre à la problématique de ce mémoire est assez complexe. En effet, reconnaître deux types d'activités avec des fenêtrages différents dans un seul algorithme est quelque chose d'assez novateur dans le domaine. Les algorithmes d'apprentissage présentés dans la section 2.2 utilisent une seule et même fenêtre de temps pour toutes les activités qu'ils doivent reconnaître. Pourtant, comme vues précédemment, les AVQ ont besoin de deux secondes d'enregistrement tandis que les exercices physiques ont besoin de six secondes, ce qui ne peut alors pas être appliqué à un seul algorithme d'apprentissage. Une solution serait alors de créer deux instances d'un algorithme, chacune entraînée avec un jeu de données différent contenant les données relatives aux AVQ ou aux exercices physiques. La solution développée utilise l'algorithme d'apprentissage des forêts d'arbres décisionnels. Étant un algorithme ayant déjà fait ses preuves dans ce domaine, il était un des choix les plus pertinents. Bien que cette solution puisse sembler simpliste, elle pose un problème majeur. Le concept même d'un algorithme d'apprentissage est de trouver une classe à un nouvel enregistrement, peu importe si elle correspond bien aux activités qu'il est censé reconnaître. Cette action rendrait l'utilisation simultanée de ces deux instances impossible, car chacune reconnaîtrait une activité de leur domaine, et il serait alors impossible de savoir quelle activité est réellement en cours de réalisation. La solution imaginée est alors d'instancier ces deux algorithmes d'apprentissage au lancement, mais de n'en utiliser qu'un seul à la fois pour reconnaître les activités courantes. L'avantage est qu'une fois entraînées, les deux instances n'ont plus besoin de se réentraîner par la suite, ce qui permet donc de sauver du temps d'apprentissage. D'ailleurs, les jeux de

données utilisés pour l'entraînement des deux instances sont des sous-parties de celui exploité au chapitre 5. De plus, étant donné que la solution s'exécute une seule fois et reconnaît des activités constamment par la suite, l'instance appliquée par défaut est celle permettant de reconnaître les AVQ. Cependant, comme mentionné dans la section 4.2, une activité a dû être ajoutée afin d'avertir le système que l'utilisateur va réaliser ses exercices physiques : la rotation du poignet. Cette activité a été ajoutée dans les AVQ pour que l'instance des exercices physiques soit utilisée à la place de celle des AVQ lors de sa détection.

En revanche, la modulation inverse est impossible de cette manière, puisque la rotation du poignet se fait de manière très rapide, et ne peut alors pas être appliquée sur une fenêtre de six secondes (qui est celle des exercices physiques). Afin de quand même satisfaire cette spécificité, un filtre a été développé pour repasser à la reconnaissance d'AVQ depuis la reconnaissance d'exercices physiques. Étant donné qu'un algorithme d'apprentissage reconnaît toujours une activité, peu importe le degré de fiabilité avec lequel il la reconnaît, ce filtre permet de gérer si un exercice physique a été réellement effectué ou non. Si ce filtre n'était pas présent, une fois la solution passée en mode de prédiction d'exercices physiques, elle ne reviendrait jamais dans le mode de prédiction des AVQ. Pour vérifier l'exactitude de la prédiction, le filtre se base sur l'attribut d'énergie préalablement expliqué. C'est l'attribut ayant la plus grande variation en fonction de l'activité effectuée. Or, les exercices physiques prescrits demandent une énergie plus importante que celle produite lors de la réalisation d'AVQ, ce qui permet de vérifier si la prédiction réalisée est cohérente ou non.

Par exemple, supposons qu'un utilisateur marche juste après avoir réalisé ses exercices. Le mode de prédiction du système est alors encore en prédiction d'exercices. Or, si un exercice physique demande une énergie avoisinant les 500, et que l'énergie produite lors de la marche est autour de 200, le filtre détectera alors une erreur comme quoi l'exercice physique détecté n'est pas cohérent avec ce qui a été réellement effectué. Avec ce filtre, si

jamais plusieurs classifications à la suite sont erronées, le système réactivera sa prédiction d'AVQ plutôt que la prédiction d'exercices physiques.

De plus, comme on le verra dans le chapitre 5, les meilleurs résultats ont été obtenus avec cet algorithme. Afin de résumer la solution développée, la figure 18 est disponible sous la forme d'un pseudo-code. Dans cette figure, le « filtre\_1 » représente le filtre détaillé ci-dessus, et le « filtre\_2 » représente celui qui combine les itérations successives du premier filtre si la prédiction courante est envers les exercices physiques, ou qui vérifie la détection de la rotation du poignet si la prédiction courante est envers les AVQ. Afin d'illustrer au mieux la solution conçue, la figure 19 représente le schéma de fonctionnement du système complet.

Avec un tel système, il est très facile de quantifier l'activité quotidienne d'un utilisateur. De plus, des améliorations sont toujours envisageables, étant donné que chaque donnée est très accessible depuis le programme. Une guidance assistée envers l'utilisateur pourrait permettre de diffuser des encouragements au moment où le système détecte un affaiblissement, ou toute autre action permettant d'augmenter le moral de l'utilisateur au moment des exercices. Il serait également possible de placer d'autres capteurs pour récolter des données supplémentaires, en fonction des nécessités de chacun. Cependant, malgré tout ce que ce système peut offrir, il reste une amélioration envisageable pour un travail futur. En effet, étant donné l'automatisation du système, il est dommage de devoir réaliser un mouvement (rotation du poignet) pour passer d'un mode à l'autre ; il serait intéressant de se demander comment ce mouvement pourrait être supprimé, afin d'offrir un automatisme encore plus poussé.



```

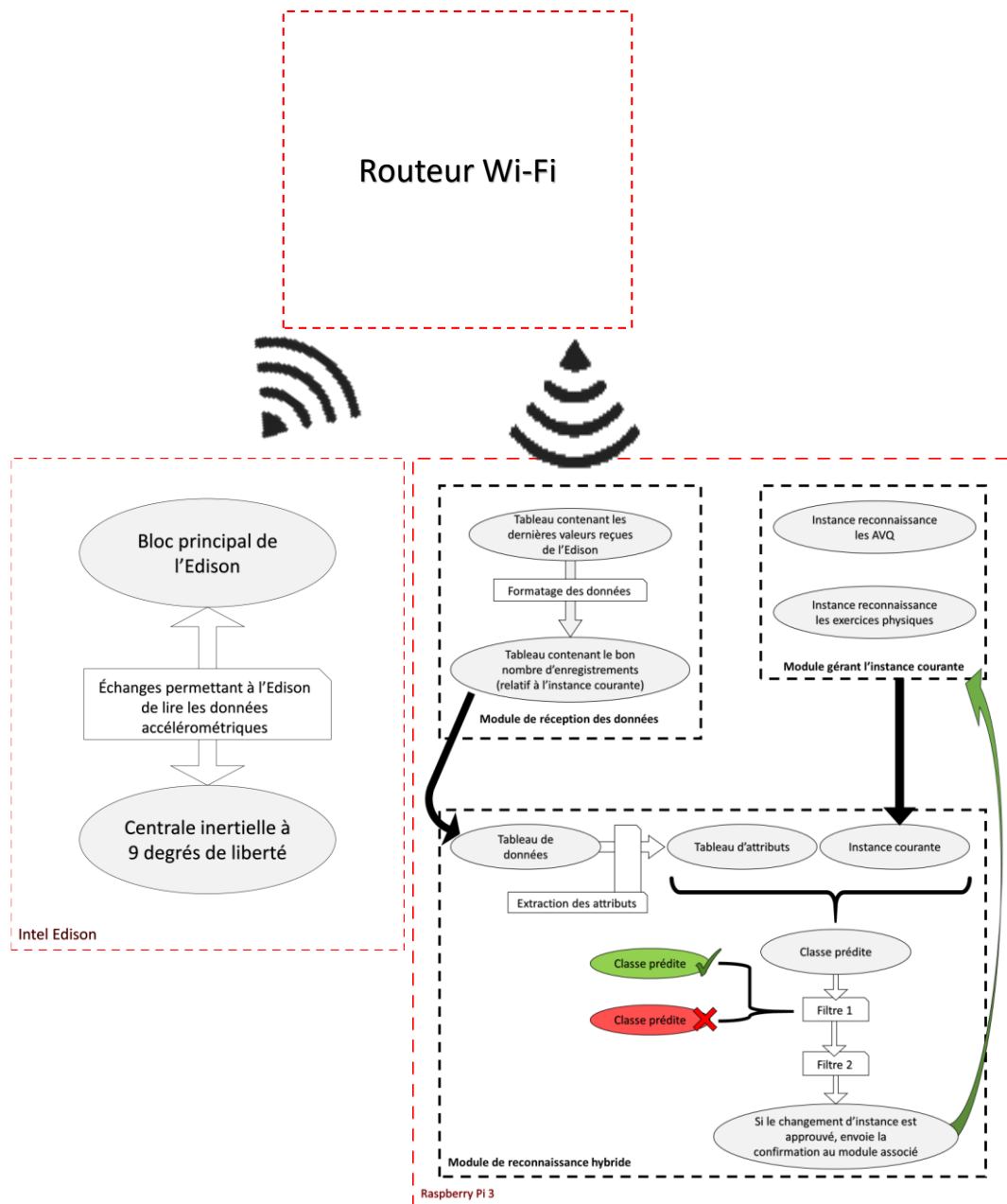
Récupérer constamment les données de l'Edison

Initialiser l'algorithme d'apprentissage reconnaissant les AVQ
Initialiser l'algorithme d'apprentissage reconnaissant les
exercices physiques

TANT_QUE le programme n'est pas manuellement coupé, FAIRE
.
. Formatter les données reçues de l'Edison,
. ou attendre s'il n'y en a pas assez
.
. Extraire les attributs du jeu de données
. Utiliser l'algorithme d'apprentissage courant pour prédire
la classe
.
. SI le filtre_1 approuve la prédiction, ALORS
.   . Stocker la prédiction
. SINON
.   . Rejeter la prédiction
. FIN_SI
.
. SI le filtre_2 approuve un changement de mode, ALORS
.   . Changer d'instance de prédiction (AVQ ou exercices)
. FIN_SI
.
FIN TANT QUE

```

**Figure 18 - Pseudo-code de la méthode de reconnaissance hybride**



**Figure 19 - Schéma représentant le fonctionnement de la méthode de reconnaissance hybride**

## CHAPITRE 5

### VALIDATION DU SYSTÈME

#### 5.1 INTRODUCTION

Afin de vérifier que la solution développée était fonctionnelle, adéquate et performante, une validation a été réalisée. De plus, les différents algorithmes d'apprentissage énoncés au chapitre 2 ont été appliqués sur un jeu de données du LIARA afin d'analyser la robustesse de la solution proposée. Les résultats obtenus lors de ces analyses sont présentés à la section 5.3. De plus, ces mêmes résultats ont été soumis à des indicateurs de performances de classification, permettant de conclure sur la performance globale de la solution proposée.

#### 5.2 STRUCTURE DU JEU DE DONNÉES

Le jeu de données utilisé pour réaliser l'ensemble des tests nécessaires à l'évaluation du système contient 13 séries de données. Elles représentent les valeurs d'une centrale inertielle (donc accélérométriques, gyroscopiques et magnétométriques) enregistrées lors de la réalisation d'AVQ ou d'exercices physiques, formatées respectivement sur une fenêtre de 2 secondes et de 6 secondes. Au total, 130 enregistrements (90 AVQ et 40 exercices physiques) composent chaque série, ce qui permet d'avoir un bon aperçu de la qualité et de la robustesse du système en utilisant ce jeu de données. Le calcul de la quantité de données pouvant être acquises en temps réel est intéressant. En effet, puisque l'Edison envoie ses données à un rythme de 100 Hz, cela résulte en 1 800 points de données lors de la réalisation d'une seule AVQ ( $1\,800 = 100 \times 9 \times 2$ , avec 100 étant la fréquence, 9 le nombre d'axes et de capteurs combinés, et 2 le temps (en secondes)) et 5 400 pour un exercice physique (même calcul que précédemment avec 6 secondes). Un entraînement normal de 10 répétitions de chaque exercice et AVQ résulterait alors en 324 000 points de données ce qui est un nombre impressionnant compte tenu de la courte durée d'un entraînement.

### 5.3 ANALYSE DES RÉSULTATS

Lors de l'analyse des résultats, un logiciel bien connu du domaine a été employé : Weka (Frank *et al.*, 2016). L'utilisation de ce logiciel a permis de tester les données sur chaque algorithme énoncé dans le chapitre 4, alors que la solution développée n'en aurait utilisé qu'un seul : celui ayant les meilleurs résultats. L'évaluation de la classification est effectuée à l'aide d'indicateurs bien connus dans le domaine du forage de données (Witten *et al.*, 2016). Ces indicateurs sont calculés à partir du nombre de vrais positifs (*VP*), de faux positifs (*FP*), de vrais négatifs (*VN*) et de faux négatifs (*FN*). Ce sont des indicateurs utilisés pour fournir des résultats facilement interprétables par la communauté (par exemple le taux de reconnaissance est un indicateur de performance). Par la suite, les résultats seront exposés sous plusieurs formes. Trois formes d'analyses ont été utilisées, et la raison de leur utilisation sera expliquée dans la section 5.3.2.

#### 5.3.1 INDICATEURS DE PERFORMANCE DE CLASSIFICATION

Pour évaluer la performance d'une classification, il est bien d'utiliser des indicateurs de performance. Celui qui est sûrement le plus connu est le taux de reconnaissance (aussi appelé l'exactitude). C'est un simple rapport entre le nombre de prédictions correctes, et le nombre total de prédictions. La formule pour le calculer est alors la suivante :

$$exactitude = \frac{VP + VN}{VP + VN + FP + FN}$$

#### Équation 30 - Calcul de l'exactitude

Cependant, il existe un paradoxe à cet indicateur, le rendant insuffisant lorsqu'il est utilisé seul. Afin de comprendre pourquoi cet indicateur ne doit pas être utilisé seul, une explication courte du paradoxe est donnée ici. Par exemple, dans le cadre d'un filtre de pourriels, soient les matrices de confusions du Tableau 6 et du Tableau 7. Le terme de « table de confusion » introduit les valeurs de prédiction d'un algorithme de classification en fonction

de chaque classe possible ; elle permet alors d'avoir un aperçu complet des classifications données par l'algorithme. Dans le Tableau 6, on voit un taux « normal » de classification de pourriels, dont l'exactitude (équation 31) s'élève à environ 73%, ce qui est très bien pour un filtre à pourriels. Cependant, dans le Tableau 7, on considère que le filtre placerait tous les courriels en tant que pourriels. Comme on peut l'observer, ce filtre possède une exactitude (équation 32) de 83%, ce qui est absolument déraisonnable étant donné qu'il classifie en tout temps pour la classe négative. Pourtant, il possède une exactitude plus élevée que son prédécesseur. On parle alors d'un raisonnement absurde, et cela vient confirmer le paradoxe de l'exactitude seule.

**Tableau 6 - Matrice de confusion pour le paradoxe de l'exactitude, partie 1**

	<i>Positifs prédits</i>	<i>Négatifs prédits</i>
<i>Classe positive</i>	20 (VP)	30 (FN)
<i>Classe négative</i>	50 (FP)	200 (VN)

**Tableau 7 - Matrice de confusion pour le paradoxe de l'exactitude, partie 2**

	<i>Positifs prédits</i>	<i>Négatifs prédits</i>
<i>Classe positive</i>	0 (VP)	50 (FN)
<i>Classe négative</i>	0 (FP)	250 (VN)

$$exactitude = \frac{VP + VN}{VP + VN + FP + FN} = \frac{20 + 200}{20 + 30 + 50 + 200} = 0,7333$$

**Équation 31 - Calcul de l'exactitude pour le paradoxe de l'exactitude, partie 1**

$$exactitude = \frac{VP + VN}{VP + VN + FP + FN} = \frac{0 + 250}{0 + 250 + 0 + 50} = 0,8333$$

**Équation 32 - Calcul de l'exactitude pour le paradoxe de l'exactitude, partie 2**

Pour remédier à ce problème d'indicateurs de performance, on peut utiliser d'autres indicateurs ajoutés à l'exactitude. Dans ce mémoire, plusieurs sont utilisés, à savoir la précision, le rappel et le F-Score (Sokolova *et al.*, 2006) dont le calcul est donné au tableau 8. Ces indicateurs ont été choisis puisqu'ils permettent de rendre très clairs les résultats obtenus. La précision donne une information rapide à propos du taux de vrais positifs par rapport à l'ensemble des positifs prédits, tandis que le rappel donne une information à propos du taux de vrais positifs par rapport aux prédictions effectuées pour la classe. Le F-Score représente une bonne valeur pour interpréter facilement la précision et le rappel en une seule valeur. De plus, leur utilisation intense dans beaucoup de travaux différents cités dans le chapitre 2 démontre bien leur performance. Les calculs afin de trouver les valeurs de ces indicateurs sont donnés dans le Tableau 8.

**Tableau 8 - Description des indicateurs de performances de classification**

<i>Nom</i>	<i>Formule</i>
Précision	$\text{Précision} = \frac{VP}{VP + FP}$ <p><b>Équation 33 - Formule pour calculer la précision</b></p>
Rappel	$\text{Rappel} = \frac{VP}{VP + FN}$ <p><b>Équation 34 - Formule pour calculer le rappel</b></p>
F-Score	$FScore = 2 * \frac{\text{Precision} \times \text{Rappel}}{\text{Precision} + \text{Rappel}}$ <p><b>Équation 35 - Formule pour calculer le F-Score</b></p>

Ces trois indicateurs, couplés à l'exactitude sont déjà très bons. Cependant, afin d'ajouter encore plus de robustesse à l'authenticité des résultats, la Kappa de Cohen (Cohen, 1968) a été ajoutée aux indicateurs utilisés. C'est un indicateur très utilisé dans différents

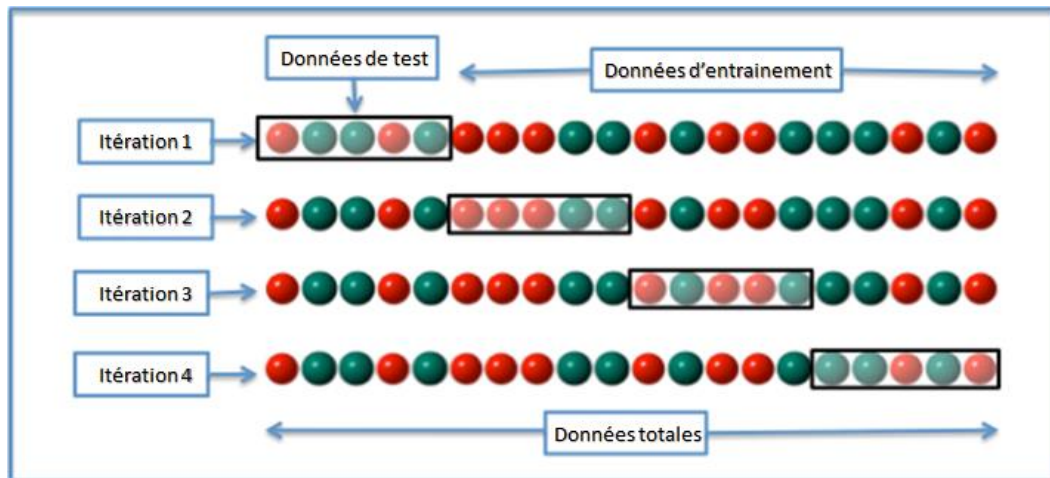
travaux (Liu *et al.*, 2016), et qui est encouragé à l'utilisation par les travaux de Ben-David (Ben-David, 2007). La définition du Kappa est observable dans l'équation 36, avec  $P_o$  et  $P_a$  étant respectivement les probabilités observées et attendues. Par l'utilisation de ces 5 indicateurs de performance, la robustesse et la confiance dans les résultats sont considérablement augmentées.

$$Kappa = \frac{P_o - P_a}{1 - P_a}$$

**Équation 36 - Formule pour calculer le Kappa**

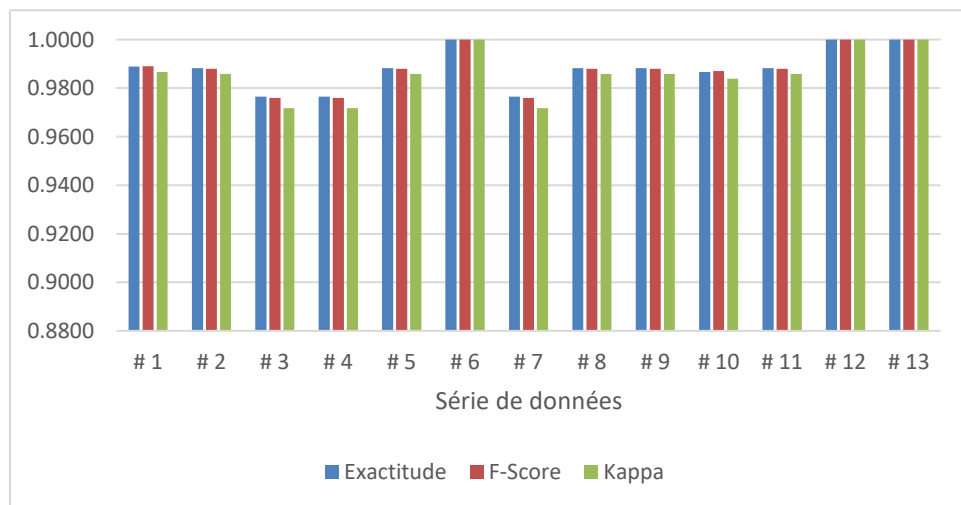
### 5.3.2 RÉSULTATS

Avec l'ensemble des éléments extraits du jeu de données utilisé, il a été possible d'effectuer plusieurs approches, afin de tester différentes hypothèses. Premièrement, l'analyse a été portée sur un entraînement avec les données d'une série de données et un test sur ces mêmes données, ce qu'on va appeler une analyse « propre ». Ensuite, l'analyse a été effectuée sur un entraînement avec les données de toutes les séries combinées sauf une seule, et un test avec les données de cette même série exclue, ce qu'on va appeler une analyse en *Leave-One-Out* (LOO). Pour finir, une analyse des différents algorithmes



**Figure 20 - Représentation d'une validation croisée avec  $k = 4$**

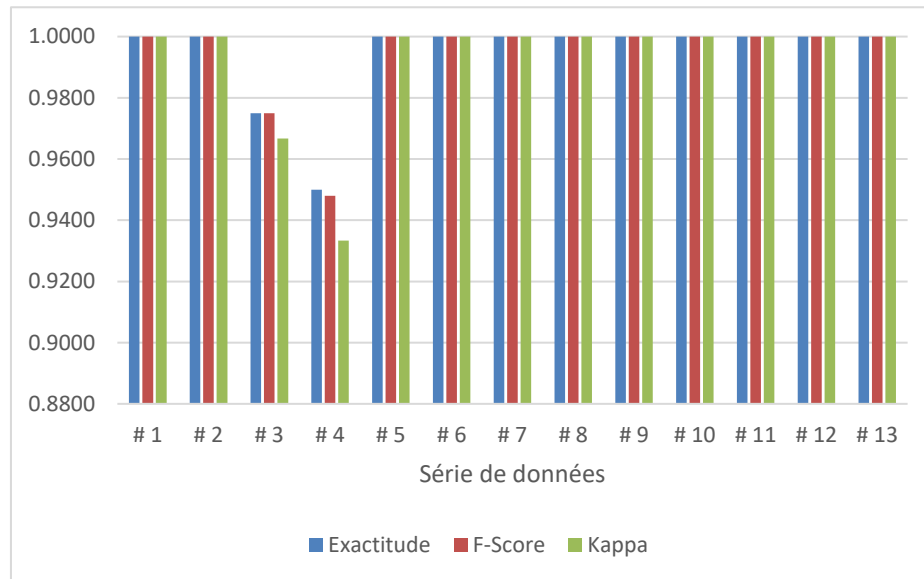
d'apprentissage a été effectuée afin de les comparer entre eux. Dans l'ensemble des graphiques de cette section, les résultats seront démontrés grâce à trois indicateurs discutés à la section 5.3.1 précédente. Malgré que cinq indicateurs étaient observés, seuls trois indicateurs sont retenus, à savoir l'exactitude, le F-Score et le Kappa. La précision et le rappel sont exclus étant donné qu'ils interviennent lors du calcul de la F-Score (voir équation 35). De plus, afin de ne pas entraîner sur des données et tester sur ces mêmes données, la validation croisée est utilisée. Le principe de cette validation est relativement simple et permet d'avoir un nombre de tests importants, même si la quantité de données est faible. Cette validation se repose sur un nombre  $k$ , représentant un nombre entier supérieur à 1. En règle générale, il est utilisé avec pour valeur  $k = 10$ , comme l'indique la littérature (Refaeilzadeh *et al.*, 2009). Il sépare alors le jeu de données en 10 parties égales, s'entraîne sur 9 parties, et teste sur la partie restante. Ensuite, il réitère ce fonctionnement avec 9 différentes parties, et ce jusqu'à avoir testé chaque combinaison possible. Pour finir, les indicateurs finaux correspondent à la moyenne obtenue pour chaque itération. Un exemple visuel de cette validation croisée est donné dans la figure 20, avec une valeur de  $k = 4$ .



**Figure 21 - Résultats propres pour les AVQ (forêts d'arbres décisionnels)**



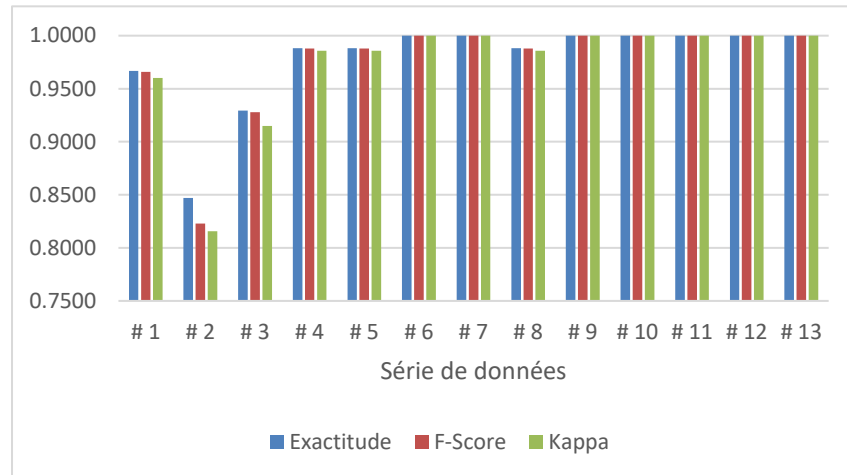
Pour commencer l'analyse des résultats, l'algorithme des forêts d'arbres décisionnels a été entraîné avec les données d'une série, et testé sur ces mêmes données par validation croisée. Les deux graphiques disponibles à la figure 21 et à la figure 22 représentent respectivement les résultats propres pour les AVQ et pour les exercices physiques.



**Figure 22 - Résultats propres pour les exercices physiques (forêts d'arbres décisionnels)**

Comme on peut le voir, les résultats sont excellents. Lorsqu'on analyse les données des AVQ (figure 21), on peut observer une plus grande disparité dans les résultats que lors de l'analyse des données des exercices physiques (figure 22). Dans l'analyse des AVQ, on a une Kappa située entre 97,17% et 100% ce qui est impressionnant (à titre de comparaison, les travaux de (Bao, Intille, 2004) avoisinent une exactitude de 84% pour une exactitude de 98,82% dans notre reconnaissance d'AVQ). De plus, on remarque aussi une très bonne statistique Kappa pour les exercices physiques, la valeur de Kappa se situant entre 93,33% et 100%.

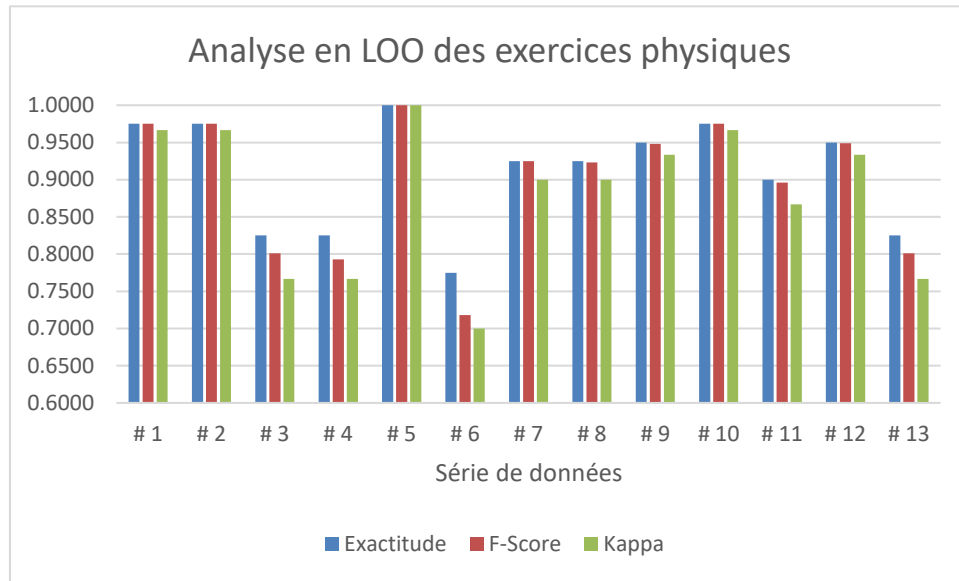
Par ailleurs, une autre hypothèse en cas réel peut être évoquée. Serait-ce possible de ne pas réaliser un entraînement de façon personnalisé du système si la banque de données d'entraînement est suffisamment conséquente ? Afin de pouvoir répondre à cette hypothèse,



**Figure 23 - Analyse en LOO des AVQ (forêts d'arbres décisionnels)**

il a fallu opter pour un autre mode d'analyse : le LOO. Pour réaliser cette analyse, il a alors fallu entraîner l'algorithme d'apprentissage sur les données de 12 séries de données, et le tester sur la dernière restante. Les données de la 13<sup>e</sup> série n'étant pas dans l'ensemble d'apprentissage de l'algorithme, l'hypothèse posée correspond tout à fait. Les deux graphiques présentés à la figure 23 et à la figure 24 montrent respectivement les résultats obtenus en LOO des AVQ et celle des exercices physiques.

Les résultats donnés à la figure 23 pour les AVQ sont bien meilleurs que les résultats présentés précédemment à la figure 21. On peut donc déduire ici qu'il serait mieux pour la reconnaissance d'AVQ d'avoir un grand jeu de données pour entraîner l'algorithme, plutôt que d'entraîner l'algorithme avec une seule et unique série de données. Cependant, les résultats obtenus en LOO des exercices physiques (figure 24) sont moins bons que ceux obtenus lors de l'analyse propre (voir figure 22) d'environ 11%. Ce qu'il est possible de déduire de ces résultats est qu'en ce qui concerne les exercices physiques, l'algorithme doit être entraîné avec une seule série de données pour un taux de reconnaissance plus efficace. Le taux de reconnaissance varierait beaucoup moins (de 70% à 100% pour l'analyse en LOO, à un intervalle entre 93% et 100% pour une analyse propre).

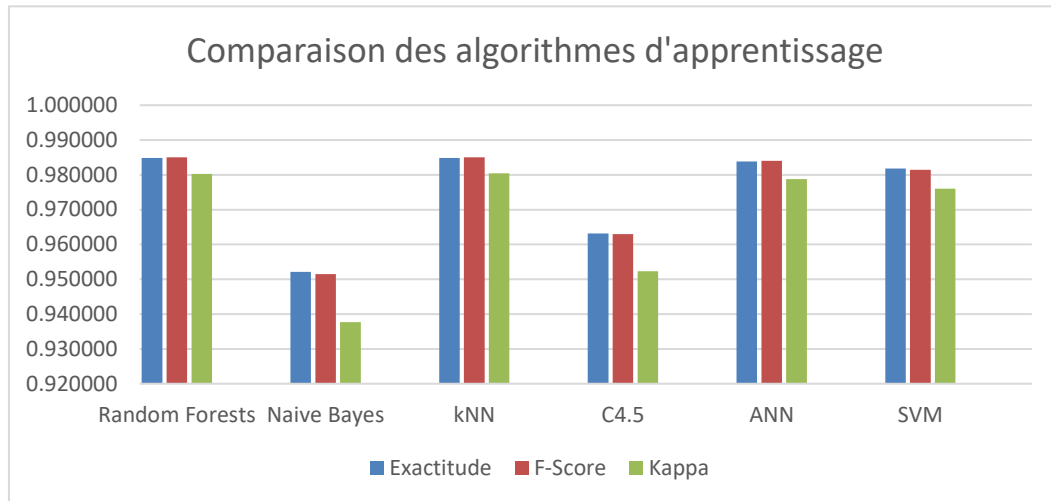


**Figure 24 - Analyse en LOO des exercices physiques (forêts d'arbres décisionnels)**

Finalement, il reste à évaluer quel algorithme d'apprentissage s'avère le meilleur choix dans notre situation. Pour ce faire, tous les éléments des séries de données ont été mis en commun afin de pouvoir tester chaque algorithme énoncé à la section 2.2. Un algorithme spécifique de chaque catégorie d'algorithme a été choisi pour la représenter. Le Tableau 9 suivant permet alors de faire l'association entre chaque algorithme d'apprentissage utilisé et la catégorie qu'il représente. De plus, le graphique illustrant les résultats obtenus est présenté à la figure 25.

**Tableau 9 - Catégories d'algorithme et leur représentant**

<i>Catégorie</i>	<i>Algorithme utilisé</i>
Classification naïve bayésienne	NaiveBayes
Recherche des plus proches voisins	k-NN
Arbre de décision	C4.5
Forêts d'arbres décisionnels	Random Forests
Machines à vecteurs de support	SVM Multi-Class
Réseaux de neurones artificiels	ANN



**Figure 25 - Comparaison des algorithmes d'apprentissage**

Dans l'ensemble, chaque algorithme utilisé est très bon. En effet, la Kappa minimale observée est celle obtenue lors de l'application des algorithmes de classification naïve bayésienne. Malgré que ce soit l'algorithme le moins performant, il possède quand même une Kappa de 93,77% ce qui est excellent. On peut également observer que les deux meilleurs algorithmes d'apprentissage à utiliser dans le système sont représentés par les forêts d'arbres décisionnels et par la recherche des plus proches voisins, qui ont respectivement une Kappa de 98,03% et 98,05%. À la lumière de ces résultats, l'adoption de l'algorithme des forêts d'arbres décisionnels pour notre solution proposée est pleinement justifiée.

## 5.4 CONCLUSION

Les résultats obtenus grâce aux analyses des séries de données dont on disposait permettent de dire que la solution proposée est très performante. Les meilleurs taux de reconnaissance sont obtenus avec les algorithmes d'apprentissage de la recherche des plus proches voisins et des forêts d'arbres décisionnels, qui sont tous deux très utilisés dans le domaine de la reconnaissance d'activités. Lors de la comparaison entre l'analyse propre et l'analyse LOO, un phénomène a été observé : les AVQ ont un plus haut taux de reconnaissance lorsque leur apprentissage est couplé avec plus de séries de données. En

revanche, la même observation n'est pas vérifiée avec les exercices physiques, où le taux de reconnaissance est meilleur avec un entraînement propre. Néanmoins, le système propose des taux de reconnaissance très élevés sur l'ensemble des analyses effectuées indiquant que sa performance est très haute. L'utilisation combinée d'attributs choisis pertinemment du domaine temporel et du domaine fréquentiel couplée à un haut taux de transmission a permis d'atteindre de tels taux.

## CONCLUSION

Le domaine de la reconnaissance d'activités évolue sans cesse : les algorithmes deviennent de plus en plus performants et de nouveaux attributs voient le jour. Cependant, il n'existe à ce jour aucune façon de reconnaître deux types d'activités sur deux fenêtres de temps différentes avec des données accélérométriques. La solution hybride proposée dans ce mémoire est un prototype permettant de faire fonctionner un algorithme sur deux fenêtres de temps différentes, s'adaptant au type de données qu'il doit recevoir. Dans le chapitre 2, plusieurs articles portant sur la reconnaissance d'activités ont été étudiés. Ces articles donnaient un angle de raisonnement pour répondre à la problématique de ce mémoire, par les différentes approches réalisées par d'autres chercheurs. Le choix final du prototype s'est porté sur l'utilisation d'un bracelet composé d'une centrale inertielle à 9 degrés de liberté. Cependant, dans le chapitre 3, lors de la présentation complète du système développé, une amélioration du système a été suggérée. Dépassant largement le cadre de ce mémoire, cette amélioration n'a pas encore été implémentée, mais le sera très prochainement par l'équipe du LIARA. Dans le chapitre 4, la méthode de reconnaissance hybride imaginée et développée a été présentée, permettant d'instancier deux algorithmes avec deux jeux de données différents en fonction du type d'activité à reconnaître (AVQ ou exercice physique). Les résultats obtenus lors des essais menés via un jeu de données disponible au LIARA, qui ont été présentés et discutés dans le chapitre 5, se sont révélés excellents. Ils démontrent que les meilleurs taux de reconnaissance sont obtenus par les deux algorithmes de forage de données suivants : les forêts d'arbres décisionnels et les k plus proches voisins. De plus, le choix des attributs était également très bon, puisque l'ensemble des algorithmes étudiés dans ce mémoire donnait de très bonnes performances au niveau des différents indicateurs.

Le système a été réalisé afin de répondre à la problématique donnée. Cependant, le développement personnalisé de ce système permet d'avoir accès à beaucoup plus de données que d'autres systèmes utilisant un logiciel existant. Avoir contrôlé l'entièreté du développement

logiciel a permis de laisser possible l'amélioration du projet. En effet, chaque partie de code est indépendante, telle que la réception des données, l'extraction d'attributs, ou l'algorithme d'apprentissage en lui-même. Cette méthode de développement rend possible l'ajout de capteurs ou d'effecteurs au système afin d'ajouter des fonctionnalités. Par exemple, avec l'ajout de haut-parleurs, il serait possible d'effectuer une assistance ou de la guidance avec ce système. L'utilisateur aurait alors un système chez lui qui pourrait l'encourager lors de la réalisation de ses exercices, ou qui pourrait même l'encourager à réaliser son programme d'entraînement s'il n'a pas encore été fait. La méthode employée ici permet alors l'ajout de toutes sortes de fonctionnalités.

Étant sujet à améliorations, ce système pourrait subir des mises à jour dans de futurs travaux. En effet, il a été observé à la section 3.4 que le bracelet avait quelques soucis d'autonomie, et pourrait alors être modifié afin qu'il puisse garder sa charge plus longtemps. De plus, comme mentionné dans le paragraphe précédent, la modulation effectuée dans la programmation du Raspberry permet d'ajouter autant de modules que voulu sans complication. Tout ce qui a été réalisé lors de ce mémoire peut alors être amélioré dans de futurs travaux. Une fois les améliorations effectuées, des expérimentations sur des patients atteints de DM1 pourront être effectuées.

Finalement, comme évaluation personnelle, je dirais que ce projet de maîtrise a été un projet intéressant qui m'a permis d'allier plusieurs domaines de l'informatique et d'en découvrir d'autres afin d'arriver à la solution finale. J'ai particulièrement appris dans le domaine du forage de données et de l'intelligence artificielle, pour réaliser de la reconnaissance d'activités en temps réel. De plus, cette expérience m'a permis de développer des compétences importantes dans le monde de la recherche, comme sa méthodologie, l'écriture, et d'apprécier l'ambiance générale de cet univers. Également, comme mentionné plus haut dans ce mémoire, ce projet m'a permis de produire un article scientifique paru dans la conférence « *The 14th IEEE International Conference on Ubiquitous Intelligence and Computing* », que je suis allé

personnellement présenter à San Francisco. De telles expériences ne pouvaient avoir comme conclusion que de me faire poursuivre dans le monde de la recherche lors d'études doctorales.



## RÉFÉRENCES

Abrahamsson, P. *et al.* (2013) 'Affordable and energy-efficient cloud computing clusters: The Bolzano Raspberry Pi cloud cluster experiment', *Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom*, 2, pp. 170–175. doi: 10.1109/CloudCom.2013.121.

Adafruit (2015) *Flora*. Disponible à: <https://www.adafruit.com/product/659> (Consulté: 12 Octobre 2017).

Bao, L. & Intille, S. S. (2004) 'Activity Recognition from User-Annotated Acceleration Data', *Pervasive Computing*, pp. 1–17.

Baum, E. B. (1988) 'On the capabilities of multilayer perceptrons', *Journal of Complexity*, 4(3), pp. 193–215. doi: 10.1016/0885-064X(88)90020-9.

Bayat, A. *et al.* (2014) 'A study on human activity recognition using accelerometer data from smartphones', *Procedia Computer Science*. Elsevier Masson SAS, 34(C), pp. 450–457. doi: 10.1016/j.procs.2014.07.009.

Bayes, M. & Price, M. (1763) 'An Essay Towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S.', *Philosophical Transactions of the Royal Society of London*, 53(0), pp. 370–418. doi: 10.1098/rstl.1763.0053.

Ben-David, A. (2007) 'A lot of randomness is hiding in accuracy', *Engineering Applications of Artificial Intelligence*, 20(7), pp. 875–885. doi: 10.1016/j.engappai.2007.01.001.

Bieber, G. *et al.* (2010) 'Mobile physical activity recognition of stand-up and sit-down transitions for user behavior analysis', *Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments - PETRA '10*, p. 1. doi: 10.1145/1839294.1839354.

Bishop, Y. M. M. *et al.* (1975) 'Discrete multivariate analysis: Theory and practice.', *Cambridge, Massachusetts: Massachusetts Institute of Technology Press Google Scholar*.

Bouchard, G. *et al.* (1989) 'Origin and diffusion of the myotonic dystrophy gene in the Saguenay region (Quebec).', *The Canadian journal of neurological sciences. Le journal canadien des sciences neurologiques*, 16(1), pp. 119–22. doi: 10.1017/S0317167100028651.

Breiman, L. *et al.* (1984) *Classification and Regression Trees*. Taylor & Francis (The Wadsworth and Brooks-Cole statistics-probability series). Disponible à:

<https://books.google.ca/books?id=JwQx-WOmSyQC>.

Breiman, L. (1999) 'Random Forests', *Machine Learning*, 45(5), pp. 1–35. doi: 10.1023/A:1010933404324.

Brigham, E. O. & Morrow, R. E. (1967) 'The fast Fourier transform', *IEEE Spectrum*, 4(12), pp. 63–70. doi: 10.1109/MSPEC.1967.5217220.

CEFRIQ (2015) *Mobilité au Québec: la croissance se poursuit*. Disponible à: <http://www.cefrio.qc.ca/netendances/mobilite-quebec-la-croissance-se-poursuit/telephone-intelligent-et-tablette-poursuivent-essor/#progression-soutenue-du-telephone-intelligent-et-de-la-tablette-numerique-en-2014> (Consulté: 29 June 2017).

Chapron, K. *et al.* (2017) 'Transportable and Scalable System for Activities and Exercises Recognition in Real-Time', in *14th IEEE International Conference on Ubiquitous Intelligence and Computing*. San Francisco, p. 7.

Cleland, I. *et al.* (2013) 'Optimal placement of accelerometers for the detection of everyday activities.', *Sensors (Basel, Switzerland)*, 13(7), pp. 9183–9200. doi: 10.3390/s130709183.

Cohen, J. (1968) 'Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit.', *Psychological bulletin*, 70(4), pp. 213–220. doi: 10.1037/h0026256.

Dencœux, T. (1995) 'A k -nearest neighbor classification rule based on Dempster-Shafer theory', *IEEE Transactions on Systems, Man, and Cybernetics*, 25(5), pp. 804–813.

Ellis, K. *et al.* (2014) 'Multi-sensor physical activity recognition in free-living', *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication - UbiComp '14 Adjunct*, pp. 431–440. doi: 10.1145/2638728.2641673.

Ferdoush, S. & Li, X. (2014) 'Wireless sensor network system design using Raspberry Pi and Arduino for environmental monitoring applications', *Procedia Computer Science*. Elsevier Masson SAS, 34, pp. 103–110. doi: 10.1016/j.procs.2014.07.059.

Frank, E. *et al.* (2016) 'The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques";', *Morgan Kaufmann, Fourth Edition*.

Gagnon, C. *et al.* (2008) 'Predictors of Disrupted Social Participation in Myotonic Dystrophy Type 1', *Archives of Physical Medicine and Rehabilitation*, 89(7), pp. 1246–1255. doi: 10.1016/j.apmr.2007.10.049.

Gallais, B. *et al.* (2015) 'Prevalence and correlates of apathy in myotonic dystrophy type 1', *BMC neurology*. BMC Neurology, 15(1), p. 148. doi: 10.1186/s12883-015-0401-6.

Gao, L. *et al.* (2014) 'Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems', *Medical Engineering and Physics*. Institute of Physics and Engineering in Medicine, 36(6), pp. 779–785. doi: 10.1016/j.medengphy.2014.02.012.

Gupta, P. & Dallas, T. (2014) 'Feature selection and activity recognition system using a single triaxial accelerometer', *Biomedical Engineering, IEEE Transactions on*, 61(6), pp. 1780–1786.

Intel (2014) *Intel® Edison Compute Module*. Disponible à: <https://software.intel.com/en-us/iot/hardware/edison> (Consulté: 6 Juillet 2017).

Jolliffe, I. T. (2002) 'Principal Component Analysis, Second Edition', *Encyclopedia of Statistics in Behavioral Science*, 30(3), p. 487. doi: 10.2307/1270093.

Khan, A. M. *et al.* (2010) 'A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer', *IEEE Transactions on Information Technology in Biomedicine*, 14(5), pp. 1166–1172.

Krasotkina, O. & Mottl, V. (2015) 'Machine Learning and Data Mining in Pattern Recognition', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9166(July 2012), pp. 425–437. doi: 10.1007/978-3-319-21024-7.

Krishnan, N. C. & Cook, D. J. (2014) 'Activity recognition on streaming sensor data', *Pervasive and Mobile Computing*, 10(PART B). doi: 10.1016/j.pmcj.2012.07.003.

Kwapisz, J. R. *et al.* (2011) 'Activity Recognition Using Cell Phone Accelerometers', *SIGKDD Explor. Newsl.*, 12(2), pp. 74–82. doi: 10.1145/1964897.1964918.

Laerhoven, K. Van *et al.* (2012) 'Multi-sensor context aware clothing', *Proceedings of the 6th IEEE International Symposium on Wearable Computers*, p. 1167218.

Lee, J. *et al.* (2007) 'A Comparative Study of Wireless Protocols', *IECON Proceedings (Industrial Electronics Conference)*, pp. 46–51. doi: 10.1109/IECON.2007.4460126.

Liu, L. *et al.* (2016) 'Smart homes and home health monitoring technologies for older adults: A systematic review', *International Journal of Medical Informatics*. Elsevier Ireland Ltd, 91, pp. 44–59. doi: 10.1016/j.ijmedinf.2016.04.007.

Mannini, A. *et al.* (2013) 'Activity recognition using a single accelerometer placed at the wrist or ankle', *Medicine and Science in Sports and Exercise*, 45(11), pp. 2193–2203. doi: 10.1249/MSS.0b013e31829736d6.

Mathieu, J. *et al.* (2003) 'Quantitative motor assessment in myotonic dystrophy.', *The Canadian journal of neurological sciences. Le journal canadien des sciences neurologiques*, 30(2), pp. 129–36. Disponible à: <http://www.ncbi.nlm.nih.gov/pubmed/12774952>.

Maurer, U. *et al.* (2006) 'Activity recognition and monitoring using multiple sensors on different body positions', *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*, pp. 4–7. doi: 10.1109/BSN.2006.6.

McCulloch, W. S. & Pitts, W. (1943) 'A logical calculus of the ideas immanent in nervous activity', *The Bulletin of Mathematical Biophysics*, 5(4), pp. 115–133. doi: 10.1007/BF02478259.

NordicSemi (2016) *nRF52832*. Disponible à: <https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF52832> (Consulté: 12 Octobre 2017).

Quinlan, J. R. (1986) 'Induction of Decision Trees', *Machine Learning*, 1(1), pp. 81–106. doi: 10.1023/A:1022643204877.

Rabiner, L. & Juang, B. (1986) 'An introduction to hidden Markov models', *IEEE ASSP Magazine*, 3(January), p. Appendix 3A. doi: 10.1109/MASSP.1986.1165342.

Raspberry Pi Foundation (2016) *Raspberry Pi 3 Model B*. Disponible à: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> (Consulté: 6 Juillet 2017).

Ravi, N. *et al.* (2005) 'Activity recognition from accelerometer data', *Proceedings of the national ...*, pp. 1541–1546. doi: 10.1007/978-3-642-02481-8\_120.

Refaeilzadeh, P. *et al.* (2009) 'Cross-Validation', in LIU, L. and ÖZSU, M. T. (eds) *Encyclopedia of Database Systems*. Boston, MA: Springer US, pp. 532–538. doi: 10.1007/978-0-387-39940-9\_565.

Shannon, C. E. (1948) 'A mathematical theory of communication', *The Bell System Technical Journal*, 27(July 1928), pp. 379–423. doi: 10.1145/584091.584093.

Shoaib, M. *et al.* (2013) 'Towards Physical Activity Recognition Using Smartphone Sensors', *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing*, pp. 80–87. doi: 10.1109/UIC-ATC.2013.43.

Sokolova, M. *et al.* (2006) 'Beyond accuracy, F-Score and ROC: A family of discriminant measures for performance evaluation', *Advances in Artificial Intelligence*, 4304, pp. 1015–1021. doi: 10.1007/11941439\_114.

ST-Microelectronics (2013a) 'LSM9DS0', (August), pp. 1–74.

ST-Microelectronics (2013b) 'LSM9DS1', (August), pp. 1–74.

Todorovski, L. & Dzeroski, S. (2003) 'Combining classifiers with meta decision trees', *Machine Learning*, 50(3), pp. 223–249. doi: 10.1023/A:1021709817809.

Tso, F. P. *et al.* (2013) 'The Glasgow raspberry Pi cloud: A scale model for cloud computing infrastructures', *Proceedings - International Conference on Distributed Computing Systems*, pp. 108–112. doi: 10.1109/ICDCSW.2013.25.

Vapnik, V. N. (1999) 'An overview of statistical learning theory.', *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 10(5), pp. 988–99. doi: 10.1109/72.788640.

Vujovic, V. & Maksimovic, M. (2015) 'Raspberry Pi as a Sensor Web node for home automation', *Computers and Electrical Engineering*, 44, pp. 153–171. doi: 10.1016/j.compeleceng.2015.01.019.

Witten, I. H. (Ian H. . *et al.* (2016) *Data mining: practical machine learning tools and techniques*.